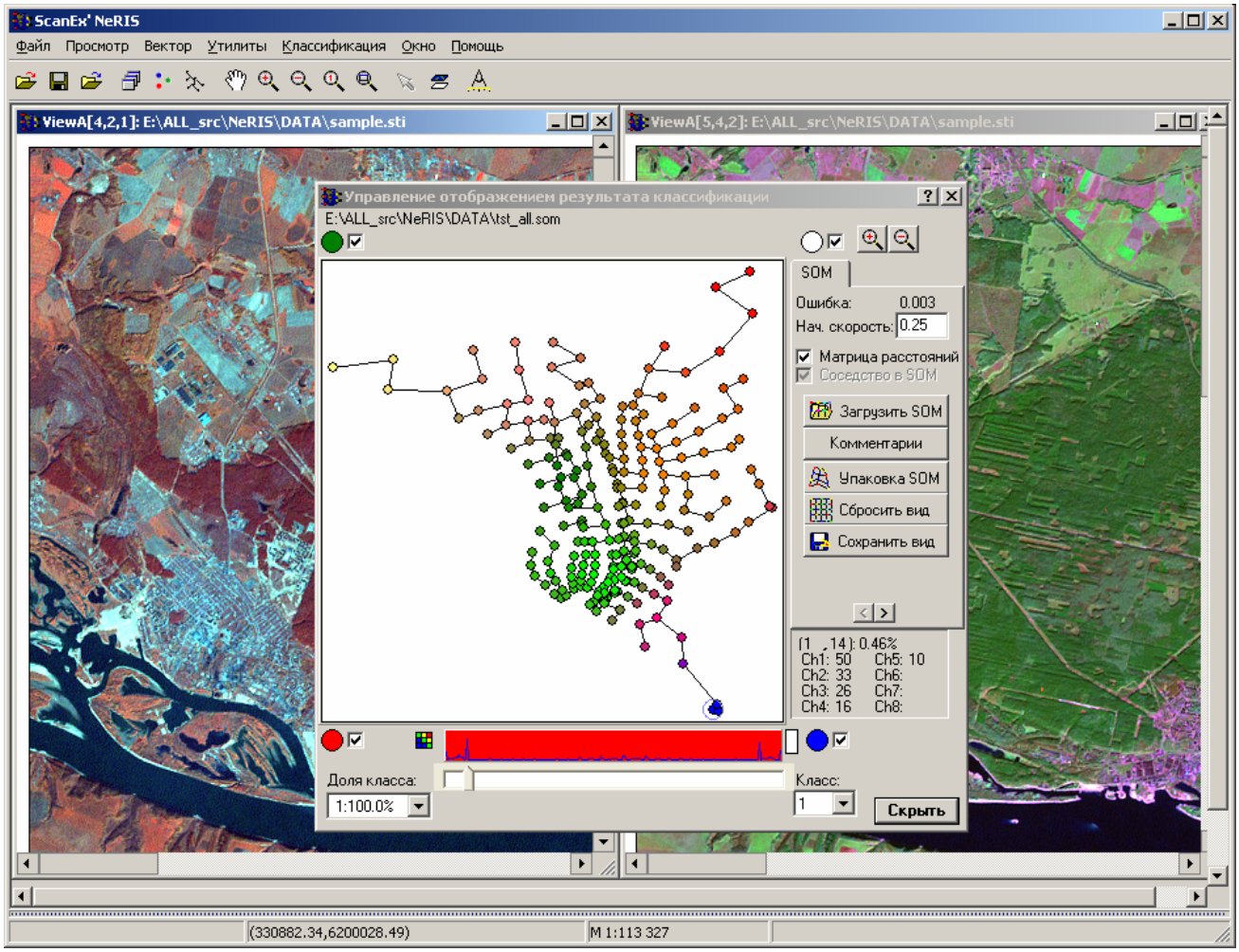
Remote sensing data thematic interpretation software

ScanEx NeRIS version 4.0
(Neural network raster interpretation system)



User's manual

Moscow 2005

NERIS SOFTWARE KEY FEATURES **4**

DATA MODELS **5**

VECTOR DATA VISUALIZATION 5
Vector cartographic information 5
RASTER DATA VISUALIZATION 6
External raster data 6
Raster data in NeRIS internal format 6
KOHONEN NEURAL NETWORKS (SELF-ORGANIZED MAPS – SOM) 7
Neural network topology 7
Distance functions 7

DATA MANIPULATIONS **8**

DISTANCE MEASUREMENTS ON MAPS 8
VECTOR DATA MANIPULATIONS 8
Creating vector objects layer 8
Loading vector objects layer 8
Layers control 8
Selecting individual objects for updating 8
Creating new vector objects 8
Deleting vector objects 9
Editing individual vector objects 9
Changing object attributive data 9
Editing vector objects drawing mode 9
RASTER DATA MANIPULATIONS IN NERIS INNER FORMAT 10
Creating primary raster 10
Loading working and secondary raster 10
Saving primary raster 10
Creating, deleting, cleaning and dubbing of raster layers 10
Creating mosaics 10
Import/export of files 10
RASTER DATA CHANGING PROCEDURES 13
Filtration 13
Principal components calculation 13
Linear analysis ..... ***Ошибка! Закладка не определена.***
Raster data view merge 13
Saving and exporting raster data 13

CLASSIFICATION USING SOM **14**

CLASSIFICATION ELEMENTS 14
Neural network training 14
Neural network calibration 14
USING SOM NEURAL NETWORKS FOR REMOTELY SENSED DATA CLASSIFICATION AND INTERPRETATION 15
Individual pixels classification using SOM 15
Image structure detection and storing 15
Hierarchical thematic classification 16
Classification post-processing using local window 16
Thematic classification of vector objects 16
Classification post-processing using Markov's random fields 16
Contextual neural networks 17

NERIS APPLICATION CONTROLS **18**

NeRIS application main dialog 18
Creating new raster file in the NeRIS inner format (Create New Raster dialog) 20
Manipulations with primary and secondary rasters (Primary Raster dialog) 22
Principle Components calculation (Principle Components dialog) 23
Using filters for image changes (Filter Plane dialog) 25
Using statistic filters (Median tab) 25
Using convolution filters (Convolution tab) 26
Calculating texture features in local window (Texture tab) 26
Classification of image "brightness relief" (Relief tab) 28

Moving data from vector maps into raster slots (Rasterize Vector Map dialog) *29*
Creating sampling text file from primary raster (Raster Sampling dialog) *30*
Creating recoding tables (Recode Thematic Plane dialog) *31*
Layers controls (Layers control dialog) *31*
Vector layers thematic display control (Legend dialog) *32*
Vector objects display mode control (MIF Display Mode dialog) *34*
Vector object data modification (Edit Object Data dialog) *34*
New vector layers creation control (Create New MIF File dialog) *36*
Raster display control (View parameters dialog) *37*
Creating maps based on neural net classification results (Create SOM Palette dialog) *42*
Hierarchy tab PopUp-menu *44*
Kohonen neural net training & calibration (Teach SOM Neural Net dialog) *45*
Classification using Kohonen neural networks (SOM Classification dialog) *50*
Contextual Kohonen neural network training and calibration (Teach Pattern SOM Neural Net dialog) *53*
Contextual Kohonen neural network classification (Classify using Pattern SOM Neural Net dialog) *56*
Classification results postprocessing using local window (SOM Classification Postprocessing dialog) *57*
Classification results postprocessing using Markov random fields (Contextual Postprocessing dialog) *59*
NeRIS APPLICATION MENU DESCRIPTION 61

NeRIS software key features

NeRIS software application (**neural raster interpretation system)** is designed for remote sensing data thematic processing and interpretation.

NeRIS enables to use both conventional (visualization, filtration, calculation to get new images) and modern methods, using adaptive algorithms based on the artificial neural networks for remote sensing input data ordination and interpretation. The latter are used for more comprehensive spatial data classification and segmentation. The main advantage of the software is in the use of artificial neural networks during the selection of thematic information and its vectorization for further use in other geo-information systems.

NeRIS allows for joint processing of vector and raster information (rasterization and vice versa vectorization for joint processing, use of vector maps to control thematic interpretation processes).

NeRIS has a big toolkit for spatial data thematic interpretation and for the creation and updating of vector digital thematic maps:

- Multi-window interface for output raster layers visualization and for the creation of views (black-white and color images composition) based on the data in the **NeRIS** internal format.
- Saving created views (together with geo-references) in formats, compatible with other geo-information systems.
- Combination of raster images with vector maps, including vector maps output in accordance with thematic legend.
- Interactive vector maps editing (creation of new digital layers, creation and deletion of objects, objects' geometric and attributive properties editing).
- Interactive mosaicking and multi-date raster creation with visual control of the results.
- Traditional images processing (use of different filters, principal components calculation).
- Training of neural networks to detect spatial structure of multi-layer raster images, visualizing the structure suitable for analysis.
- Getting new images based on the image structure, invariable with regard to a whole class of brightness transformations, which sometimes allows the user to avoid photometric correction during the multi-date images analysis.
- Multi-layer raster images segmentation using trained neural networks.
- Thematic calibration of trained neural networks in order to correlate the structure and detect its possible thematic interpretation.
- Thematic interpretation of multi-layer raster images, building thematic legends and automatic selection of thematic objects and saving them as vector maps.
- Use of post-processing of classification results both for verification and reliable thematic classification of preset vector objects.

The use of image structure detected by neural network to create invariant presentation of the image data is a unique function of the **NeRIS** software, providing great possibilities for remote sensing data thematic interpretation, not available in other software applications.

Data models

Vector data visualization

NeRIS application uses vector digital non-topological maps, designed to store geometrical and attributive data about vector map objects.

Data models visualization details are indicated below.

Vector cartographic information

It is common practice in computer cartography to present vector data in layers. The number of co-resident layers should not exceed 256. **NeRIS** application has all traditional layers' control tools (change output order when drawing; select drawing mode both for the whole layer and using thematic legend; control view and editing properties, as well as objects selection for a specific layer).

Each layer is loaded from a separate file, using **MapInfo** export format (**MIF/MID).** This format enables to save both geometric objects' description and attributive tables with different data for each of the objects; layers are saved to files in the same format.

NeRIS application uses non-topological item-by-item geo-coding for objects geometry mapping. The number of geometric primitives is conventional for non-topological systems (such as **MapInfo** or **ArcView**); **MapInfo** terminology is also used in **NeRIS** application**.** Below is the list of all geometric primitives used (relevant **MapInfo** system terms are given in brackets):

- ***Point*** – a unique point (***vertex***).
- ***Line*** – a line, represented by the start and end points (***vertices***), having the start, the end and the direction.
- ***Polyline*** – one line, represented by a sequence of points (***vertices***) in the application. The line does not have a direction; the start and end points are not selected deliberately. For the internal line presentation the word ***arc*** is used; the word ***vertex*** is used for the line points.
- ***Multiple Line*** – a group of lines regarded as one object, having one record in the ***table of attributes***. Lines inside the group are not ordered and presented in the application as mentioned above.
- ***Region*** – spatial object (region). The region is presented as a set of borders - closed rings. The border does not have selected points or selected traverse direction. For the internal border presentation the word ***closed arc*** is used; the word ***vertex*** is used for the border points. Region can be in form of a random multilinked complex polygon; the number of connected components and "islands" in each of them is not restricted.

Raster data visualization

NeRIS application allows the user to work with many types of raster images:

- Raster images in **MapInfo, GeoTiff, ERDAS Imagine 8x, ESRI BIL (*.bil, *.bsq), PCI Geomatica, ENVI, BMP, JPG** formats, as well as **ArcInfo ASCII GRID**.
- Raster images in the internal **STI (*.sti)** format of **NeRIS**, created inside **NeRIS** application after raster data processing and importing.

External raster data

NeRIS application enables to simultaneously open a random number of images, having their own windows for independent location and scale control, but with a vector data setting and output method general for all windows.

Such raster images' preview options and their use to resolve different tasks are described in details in appropriate sections of the manual.

Raster data in NeRIS internal format

Slots are used to store raster data (images) inside the application. Each of the slots contains a single ***image plane*** with 8-bit value of each pixel. Therefore, three slots are required to store one input image, displayed in True Color format (according to the number of planes in the source file).

Slots are combined in two groups, called ***primary*** and ***secondary raster***, currently with 16 slots in each. All slots of the same raster have similar size and coordinate system; coordinate systems and pixel sizes of the primary and additional raster are independent and may vary.

Both pixel coordinate system (axis from left to right and from bottom to top) and target coordinate system (Cartesian system, parallel to pixel rows and columns) with random dimensionality, normally expressed in meters, are used in the application.

From the viewpoint of image visualization (***view*** or ***presentation***), both raster are equal; whereas processing operations can be done only using primary raster (secondary raster can be used only for ***mosaicking*** from several images).

Kohonen neural networks (self-organized maps – SOM)

Neural network topology

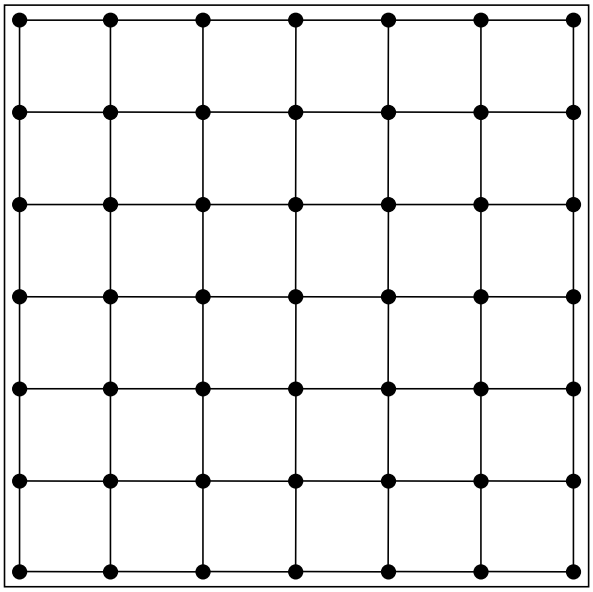
Self-organized maps, proposed by Kohonen, are single-layer neural networks with a competitive layer, where the input vector is fed at the same time into all neurons inputs. Only one winning neuron, "matching" better with the input vector, is identified based on one of the following methods to calculate distance between input vector and each neuron's weights vector.

On the other hand, we interpret neuron's weight vectors as classes' centers, defined as distance segmentation using the same distance function. In this way one-to-one correspondence between neural network neurons and their classes is defined in the initial feature space.

Neural network training implies updating of winning neuron's weight vector (shifting it in feature space towards input vector). Unlike traditional classification methods, when separate classes updating is made independently, in **SOM** network all neurons (consequently, all classes) are put into certain topological grid, specifying ***neural network topology.*** This topology has nothing to do with neuron's weight vectors position in the initial feature space and serves only to control neural network training for its self-organization. Topological grid is used to define the ***neighborhood*** of the winning neuron, which includes all adjacent network neurons in the grid, regardless of their weights distance (similarity) to the input vector.

The principal difference of SOM neural network from traditional classification systems is in the fact that besides the winning neuron, all neighboring in the neural network grid neurons are also trained which leads to neural network self-organization and determination of internal structure of multi-layer image. Determination of this very structure is the result of neural network training.

In this software version topological grid is in form of regular rectangular net, containing not more than 255 neurons. The example of a 7-by-7 set of neurons (7 rows with 7 neurons) is shown below.



Neurons numeration in neural network corresponds to their position in gridtop topology: bottom left neuron has the position (1,1), next one has the position (1,2) and so on; when saving neural network to a file the neurons are presented in rows starting with the bottom. This numeration will be further on referenced as ***coordinates in neural network grid***; in particular, such coordinates are used when calculating the neighborhood.

Distance functions

Three functions are used to define the winning neuron in the **NeRIS** application (names, used in the application, are indicated in brackets):

- Sum of absolute differences between vector components ("City Block" or "Manhattan" distance).
- Euclidean distance between vectors (Euclidean).
- Angle between vectors ("spectral angle"), calculated as relation of scalar vectors product to the product of their lengths (Spectral Angle).

All functions are selected to make neural network training stable (up to a certain limit) with respect to linear illumination changes of different image channels (planes).

Data manipulations

Data manipulations include both raster data and vector data manipulation assets.

Distance measurements on maps

Relevant toolbar hot key is used to measure distances. To measure a distance, press left mouse button over the start point and drag the cursor to the destination point. The distance is shown in the status bar at the bottom of the main widow.

Vector data manipulations

Creating vector objects layer

To create a new vector object layer, the data on its geographical projection is required; therefore it can be created only if a pre-loaded raster is available in the memory (projection parameters are taken from raster for the layer being created).

Primary raster control item or **Secondary raster control** item are used to create new layers in **Utilities** menu of the **NeRIS** master form. After selecting this menu item, a **STI** form appears, described in the corresponding section of this user's manual.

Loading vector objects layer

To load a previously created vector object layer, use L**ayers control** item in **Vector** menu of the map window or a relevant toolbar hot key. After selecting this menu item, L**ayers control** form appears, described in corresponding section of this user's manual.

Layers control

Layer control includes layers' loading, deletion, changing layers output order and mode and layer properties (visualization, editing factors, and objects selection possibilities) and saving layers to files. **Layers control** form is used for all these actions, as described in the corresponding section of this user's manual.

Selecting individual objects for updating

To select individual objects for updating, use **Select Object** item in **Vector** menu of the map window or a relevant toolbar hot key. Objects selection mode activation is indicated by changes in cursor shape and button appearance.

Click left mouse button on the object to select it (objects selection properties of the relevant layer should be activated). If there are several objects within the mouse capture area, the top layer object will be selected. If there are several top layer objects, the top one is selected (in the order of drawing). In case it is "shading" the required object, the drawing order in the layer can be changed. Select **Move down** item in the **Pop Up** menu (the selected object will now be the first in the drawing order list and will not be shading the previously drawn objects).

Only one object can be selected at a time. Hereinafter, it will be called an ***active object***.

Drawing mode, used for the active object, is specified in item **Set selection mode** in **Vector** menu of the map window.

Creating new vector objects

Editing functions should be activated in the corresponding vector layer to create a new vector object.

To create a new vector object, use **Create Object** item in **Vector** menu or **Create Object** item in **PopUp** menu of the map window. After selecting the type of the new geometric object (point, line, polyline, multiple line, region) the application activates the object creation mode; current input mode, used during object creation, is specified in **Set drawing mode** item in **Vector** menu of the map window.

Point object is created by pressing left mouse button on the required map window location.

Straight line is defined by one-at-a-time clicking with left mouse button on the start and end points of the line being created.

Polyline can be entered in two ways:

- Separate points mode – by clicking left mouse button on the map window locations where polygonal line vertices should be placed.
- Free drawing mode – by moving mouse along the future line (with depressed left mouse button and **Shift** key) which enables to outline the required objects on the raster image. After releasing the **Shift** key, the program will mode back to separate points drawing. It should be noted that in free drawing mode the polyline is simplified

down to half of pixel size (preventing to draw polygrams inside the pixel, however, removing minor details to generalize stored data size reduction).

In any case to complete polyline creation, double click with left mouse button or select **Complete Editing** item in **Pop Up** menu of the map window.

Multiple lines are originally created from one line (like in the previous item), whereas new lines are added in editing mode.

Region is originally created in form of one simple polygon similar to the polyline creation (the border is locked automatically)

Deleting vector objects

Editing functions of the corresponding layer should be activated to delete vector objects, with the active object to be deleted.

To delete a vector object, press **Del** key or select **Delete Object** item in **Vector** menu of the map window. The last deleted object can be restored using **Undelete** item in **Vector** menu of the map window.

Editing individual vector objects

Edition of geometric properties of an individual vector object includes (this type of editing is applicable to objects shown in brackets):

- Changing individual vertices' coordinates of the object (all types of objects).
- Adding new vertices and deleting existing ones (lines, polylines, multiple lines, regions).
- Adding and deleting polylines (multiple lines).
- Adding and deleting borders (regions).

Changing coordinates of individual vertices is applicable to active points, lines and polylines (all vertices are outlined with squares when displayed). For multiple lines and regions it is required first to select an active arc (it has big squares around vertices during drawing) by clicking with left mouse button on any location of the arc to be made active.

To change coordinates of the active arc vertex, press left mouse button within the outlined vertex square and drag the vertex to a new location.

To add a new vertex, press left mouse button over the line, connecting two vertices of the active arc (created vertex can be already moved to the required location).

To delete a vertex on the current arc, keep left Ctrl key depressed and click left mouse button on the vertex to be deleted.

To add a new line to a group of lines (or a new border to a region), use **Add Arc/Region** item in **Pop Up** menu that appears if the current object is a group of lines or a region. After that the program activates drawing of new line (border) to be added to object.

Changing object attributive data

To change attributive data of the selected object, use **Change Attributes** item in **Pop Up** menu of the map window. **Edit Object Data** dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

Editing vector objects drawing mode

To change the selected object drawing mode, use **Change Mode** item in **Pop Up** menu of the map window. **MIF Display Mode** dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

Raster data manipulations in NeRIS inner format

The use of raster data in **NeRIS** inner format gives the best advantages in remote sensing data processing.

Creating primary raster

To create a primary raster, use **Create Primary raster** item in **File** menu of the program window. A dialog will appear on the screen, which controls are described in details in the corresponding section of this user's manual

Loading working and secondary raster

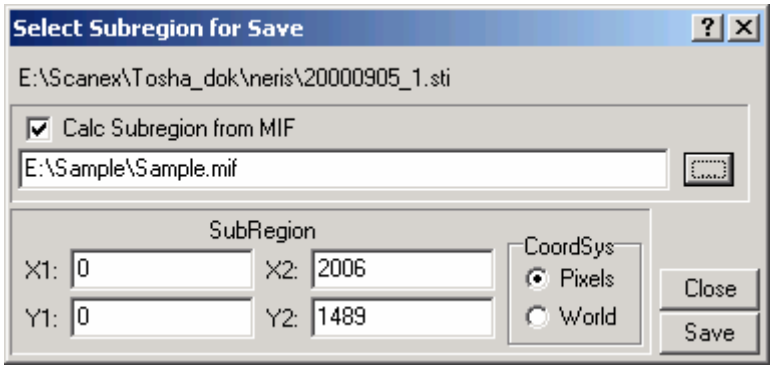
To load a working and secondary raster, use **Load Primary raster** and **Load Secondary Raster** items respectively in **File** menu of the main program window.

Saving primary raster

To save a primary raster, use **Save Primary raster** item in **File** menu of the main program window. Two options to save the primary raster are available using standard file selection dialog:

- Saving the whole raster (**Internal Raster Format** item in the **Save Dialog** list of file types).
- Saving selected subregion (**Clip Subregion in STI Format** in the list of file types of the **Save Dialog** standard window).

In the second option, **SubRegion** selection dialog appears on the screen, containing the following controls:



Checkbox **Calc Subregion from MIF.** If this control is active, the region to be saved is specified using vector map as indicated below.

Saved region borders file selection push-button enables to select the vector map file, for which the smallest rectangular, containing all objects, is calculated. This rectangular defines the raster region to be saved.

CoordSys *radio buttons group* specifies the method of saved region coordinates recording (in raster pixels or in projection coordinates).

Fields **X1, Y1** specify the left bottom angle of the saved rectangular and can be set and updated by the user.

Fields **X2, Y2** specify the right top angle of the saved rectangular and can be set and updated by the user.

Creating, deleting, cleaning and dubbing of raster layers

To create, delete, clean or back up raster layers, use **Primary raster properties** and **Secondary raster properties** items in **Utilities** menu of the main program window. A dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

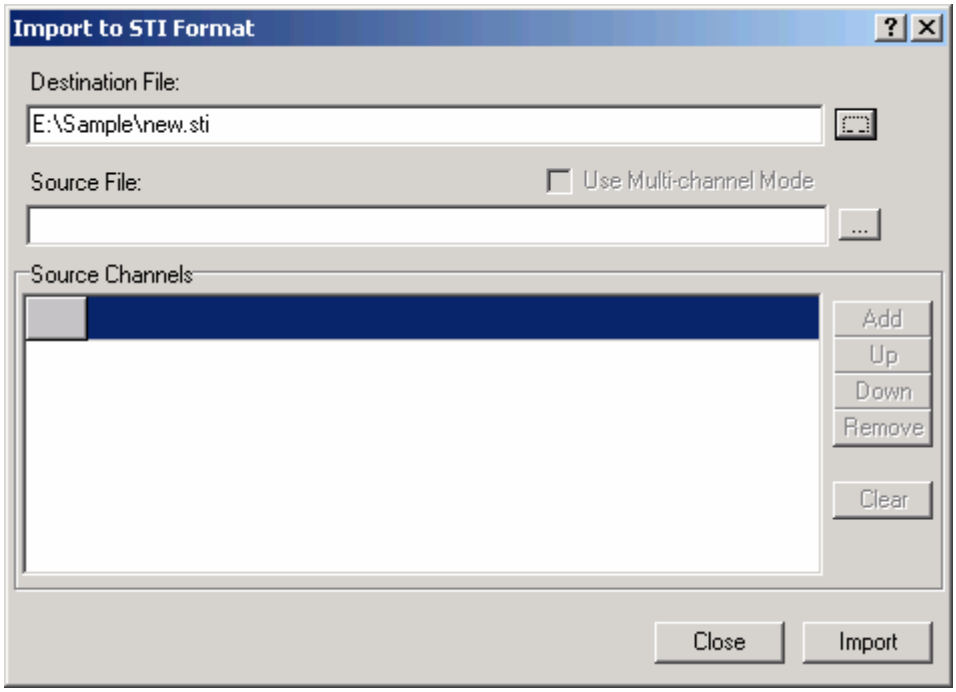
Creating mosaics

To create mosaics from several raster layers or to patch a multi-date remotely sensed data in one raster, use **Patch Primary Raster** item in **Utilities** menu of the main program window. **Patch Primary Raster** dialog appears of the screen, which controls are described in details in the corresponding section of this user's manual.

Import/export of files

Raster data importing to STI format

To work with raster data in GeoTiff, ERDAS Imagine 8x, ESRI BIL (*.bil, *.bsq), PCI Geomatica, ENVI, BMP, JPG and ArcInfo ASCII GRID (*.ASCII) formats**,** use **Import** item in **File** menu. **Import to STI format** dialog appears on the screen, containing the following controls:



Field **Destination File** specifies the output file name and format.

File selection push-button enables to choose a directory, where the new file is created.

Field **Source File** specifies input file name and format.

File selection push- button enables to select a graphical file to be imported.

Checkbox **Use Multi-channel Mode** activates the list of existing channels.

Table **Source Channels** displays the list of existing channels.

Button **Add** adds the selected file to the list of existing channels.

Button **Up/Down** moves the selected channel in the list one row up or down.

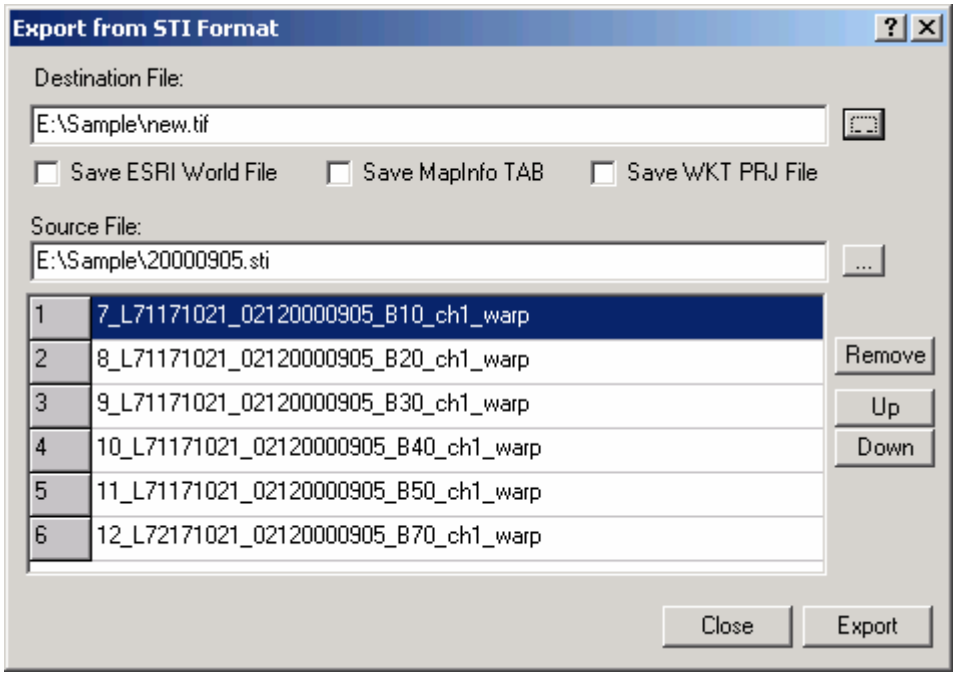
Button **Remove** deletes the selected channel from the list of existing channels.

Button **Clear** clears the list of existing channels.

Button **Import** initiates importing operation.

Exporting data from STI into other formats

For data exporting from STI format into one of the other formats (**GeoTiff, ERDAS Imagine 8x, ESRI BIL (*.bil), PCI Geomatica, ENVI, BMP, PGM, PPM и ArcInfo ASCII GRID (*.ASCII)**), use Export item in File menu. Export from STI format dialog appears on the screen, containing the following controls:



Field **Destination File** specifies output file name and format.

File selection push-button enables to choose a directory, where the new file is created.

Checkbox **Save ESRI World File**. If checked**,** World file will be saved, which is used for the creation of image spatial coordinates of the ESRI program family (ArcView, ArcGIS).

Checkbox **Save MapInfo TAB**. If checked**,** TAB file will be saved, which is used for the creation of spatial coordinates and image projection in MapInfo.

Checkbox **Save WKT PRJ File**. If checked, text file with WKT projection parameters will be saved (World known text).

Field **Source File** specifies input file name and format.

File selection push-button enables to select a graphical file to be imported.

Table **STI channels list** (located in the center of the dialog) – displays the list of existing channels.

Button **Up/Down** moves the selected channel in list one row up or down.

Button **Remove** deletes the selected channel from the list of existing channels.

Button **Export** initiates exporting operation.

Raster data changing procedures

Filtration

Filtration item in **Utilities** menu of the main program window is used to apply different filters to raster layers. **Filter Plane** dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

Principal components calculation

To calculate principal components of the given raster planes, use **Principle Components** item in **Utilities** menu of the main program window. **Principle Components** dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

Raster data view merge

To create a new map window, use **Create Working Window** item in **Window** menu of the main program window or the relevant hot key button of the menu.

To merge raster data display in the map window, use **View Parameters** item in **View** menu of the map or a relevant hot key. **View Parameters** dialog appears on the screen, which controls are described in details in the corresponding section of this user's manual.

Saving and exporting raster data

NeRIS application enables the user to save the created raster display in the map window together with the coordinates. **Save display as** item in **View** menu of the map window is used. Display settings can be saved in the following formats:

- Using one-channel **BMP** file with 256 gradations of grey, color **BMP** file with 256 colors and True Color **BMP** file (depending on the created image, coordinates are not saved).
- Saving coordinates in **TAB** format of the **MapInfo** with the option to save the image as a one-channel **BMP** file with 256 grey gradations or as a True Color **BMP** file.
- Saving the selected number of images in a standard file with bites interleaving (**BIL**), coordinates will be saved in standard files with **HDR** and **BLW** extensions.
- Saving the image and the coordinates in the inner **ScanEx** format (**STI**) with the option to save part of the image. **Save subregion** dialog is used to select the subregion, similar to primary raster saving.
- **PPM Raster** format images.

Classification using SOM

Basic methods of neural networks classification, available in **NeRIS** application, are illustrated below. Their controls are described in details in the corresponding sections of this user's manual, with the help of **Teach SOM Neural Net**, **SOM Classification, SOM Classification Postprocessin**g and **Contextual Postprocessing** dialogs.

Classification elements

Neural network training and self-organization is illustrated in this section as well as the process of its thematic calibration.

Neural network training

The purpose of neural network training is to detect the structure of the existing set of raster layers using three spaces:

- *p*-dimensional space of the source image, created by the vectors of values, taken from *p* of equally located pixels of one raster, which we call ***source image space***.
- 2-dimensional neural network coordinates space (or the coordinates within a neural network grid, acquired in Sammon mapping described below), which we call ***Sammon's space***.
- ***Thematic space,*** assigning pixels to thematic classes, having the dimension equal to the number of thematic classes.

The detection of the source image structure will be considered here as a replacement of each source image point by a certain class, represented by the vector in the same space, and the description of these classes configuration (since a certain distance function is used in training, therefore only the classes' neighborhood data can be regarded as the training result).

As we are trying to get not just a number of classes, but their structure, during the training, self-organization process is used, which locates the neighboring classes (in the source image space) within the adjacent cells of the neural network topological grids and vice versa.

To achieve self-organization during the training the neighborhood is used (within the neural network grid): after inputting the next vector of values to the neural net and defining the winning neuron, which is the closest to the input vector, it will be trained together with all the neighboring neurons.

NeRIS application is using two strategic options to choose the neighborhood (terms used in the application as indicated in brackets):

- All neurons, located not further than at the specified distance from the winning neuron, fall within the neighborhood (in neural network grid coordinates). The training rate is similar for all neurons (**Bubble**).
- All neurons of the neural network are in the neighborhood; however their training rate drops as distance from the winning neuron increases (**Gaussian**).

To bring the self-organization process to the required results, the radius of the neighborhood decreases throughout the training (for the Bubble mode), or the training rate drop expedites as the distance from the winning neuron in the grid increases (for the Gaussian mode).

Finally, for the same purpose, the training rate is reduced down to zero throughout the training, with the initial rate settings specified by the user. **NeRIS** application uses two strategies of training rate reduction (terms used in the application as indicated in brackets):

- In inverse proportion to iteration number (**Inverse Time**).
- Linear (**Linear**).

Neural network calibration

After self-organization is complete, the trained neural network detects the source image structure, which can be used for multi-layer images segmentation. However, it is more interesting to correlate the detected structure with the pixels relation to certain thematic classes.

Thematic calibration of the trained neural network is used for such correlation, by inputting vectors, corresponding to the pixels with the known thematic interpretation, to the neural network. Neural net classifies each of such vectors and remembers which thematic classes and in which proportion are represented in each neural network class. Basically, if the calibration selection is enough complete, these proportions are very close to the possibility of finding this thematic object among the pixels, related to this neural network class, consequently, the vector of such probabilities for each neural network class can be used for further creation of the classification procedures both based on the probability theory and on the fuzzy logics.

This section illustrates the principals of using trained and thematically calibrated networks for remote sensing data classification and thematic interpretation.

***Individual pixels classification using* SOM**

Classification using neural networks can be done as follows:

- To find the corresponding neural network class for each pixel (using the same standard algorithm of distance functions, as during neural network training) and to use the number of the corresponding class as the classification result.
- To use the calibration results (after finding the neural network class) to detect which thematic class the pixel is related to.

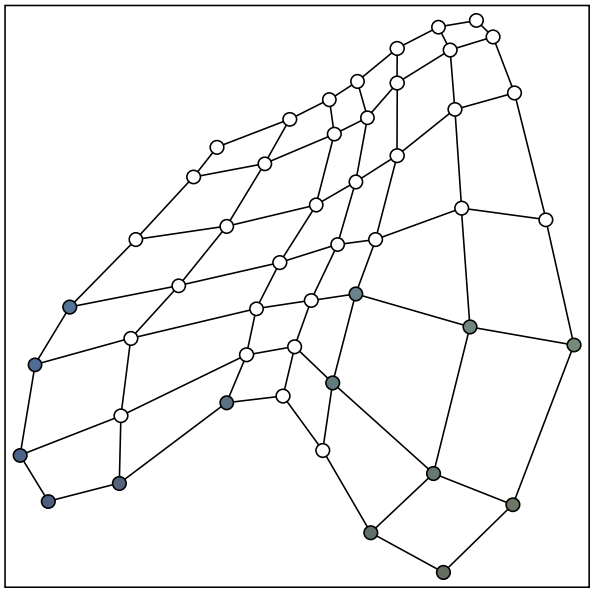
Image structure detection and storing

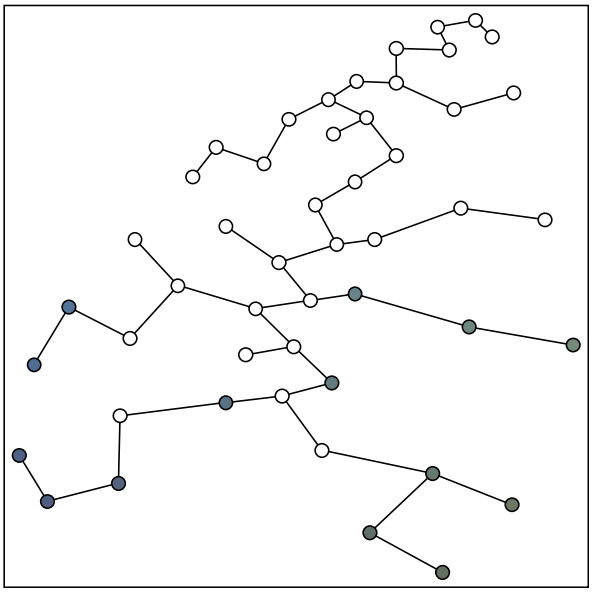
To display the detected structure of the multi-layer image after training, **NeRIS** application is using the Sammon's mapping, when the classes in the initial *p*-dimensional space are mapped in the planar grid so that the distances between the classes characters are the same as between the classes themselves. After normalizing received Sammon's coordinates, we obtain the invariant structure of the initial image (in terms of distance functions used). It's easy to see that such structure is invariant with respect to the linear transformations of the values of individual initial image planes, provided this transformation is not putting the values beyond the permissible value range.

Therefore, if we are using Sammon's coordinates for the creation of two new image planes, they keep all the data about the initial image structure. In this way, the trained and thematically calibrated neural network, which coordinate classes have been transformed into Sammon's space, can be used for thematic interpretation of new images, obtained under different atmospheric conditions (provided the image is also transformed into the Sammon's space).

Besides, Sammon's coordinates can be also used as the compression of the initial image when it is transferred to different classification procedures.

NeRIS application provides for the assessment of built classification quality and compliance of neural network structure, transformed into Sammon's space, with the initial neural network structure. Backbone tree is built for this purpose (using both the entire distance matrix between classes and the distance within the neurons grid), which is displayed over the built Sammon's map (below is the Sammon's map for trained neural network and the backbone tree image, allowing for a better understanding of the classes structure).





The absence of self-intersection of the built tree indicates a successful self-organization of the neural network and the possibility to use Sammon's coordinates for further processing.

Hierarchical thematic classification

NeRIS application provides for a wide range of remotely sensed data analysis using the multi-layer image structure, detected by the neural network.

First of all, these are the unique means of palette creation to visualize the built classification, based on the color gradient display of the image structure.

To create such "structural" palette the application is using the "color sprout" metaphor. Color sprouts – are the colors, assigned to individual neurons of the neural net by the user. To select these neurons, the user can apply the calibration results (select neural net classes, matching better with thematic classes), distribution of neural net classes throughout the territory, visually assessed when building raster images, or select them after visual structure analysis (presented as Sammon's map) for better structure visualization.

The color for the visualization of the rest of the classes is generated automatically by interpolation of color sprouts using Sammon's coordinates. As a result, the closer are the classes in Sammon's map (consequently, within the source image space as well), the closer is their color in palette. However, even the colors of neighboring classes are somewhat different, which enables to visually structure the image and detect its spatial structure. Whereas random color selection, traditionally used to visualize classification results, brings to a "fractal" picture (with a big number of classes), which losses spatial objects, the approach used in the **NeRIS** application enables to present the results of classification into a big number of classes so that there are no visual distortions of spatial structure.

Notably, there is a possibility of using different colors and its tones to build hierarchical structure, created by the neural net classification, as well as the possibility to build color legends using the neural net calibration results, when neighboring color sprouts are used for the far off in the neural net, but thematically close classes.

The possibility of building hierarchical classification using several neural networks should also be noted. The first net is used for rough thematic distribution (e.g. water objects, plants, etc.) with further selection of these classes for a detailed classification and study.

Classification post-processing using local window

As individual pixels classification, being contextually independent, does not allow for detection of a number of traditional decoding features (e.g. texture), **NeRIS** application provides for the post-processing of the already built classification using classes distribution (inside a local window), detected by the trained neural net. Since such distribution is unidimensional, a standard approach of signature selection is used to operate it (typical for the given thematic class of neural net classes' distribution). The signatures are automatically withdrawn from the trained and thematically calibrated neural network.

To verify the signatures against the required ones, the **NeRIS** application uses 8 signature matching measures, described in details in the corresponding section of this user's manual. As a result of such post-processing the thematic class, which signature is the closest to the one, calculated within the local window, is assigned to the central window pixel.

Thematic classification of vector objects

Besides the aforementioned procedure of signature calculation within the local window, the object-by-object classification procedure becomes very popular lately, when for each vector object on a certain digital vector map it is known in advance that it belongs to one thematic class. In this case, it's enough to calculate the classes' signature for classification within the limits of the object and to find the closest thematic class in the calibrated neural network. This very approach is used in the **NeRIS** application.

Classification post-processing using Markov's random fields

To get the thematic classification, "consistent" with the entire image, the application uses the Markov's random field tool, enabling to receive generalized thematic classification with the preset extent of details based on the thematic calibration of neural network.

The received classification will be coordinated not only with thematic distribution of probabilities, kept within the neural net, but also with thematic classes of this pixel "neighbors", using Gibbs energy minimization when applying Markov's random field of classes.

"Consistence" in this particular case indicates that the pixel thematic class is "inclined" to the neighboring thematic classes as a result of local changes in probability distribution of thematic classes for the given neural net class, with the change rate controlled by the user.

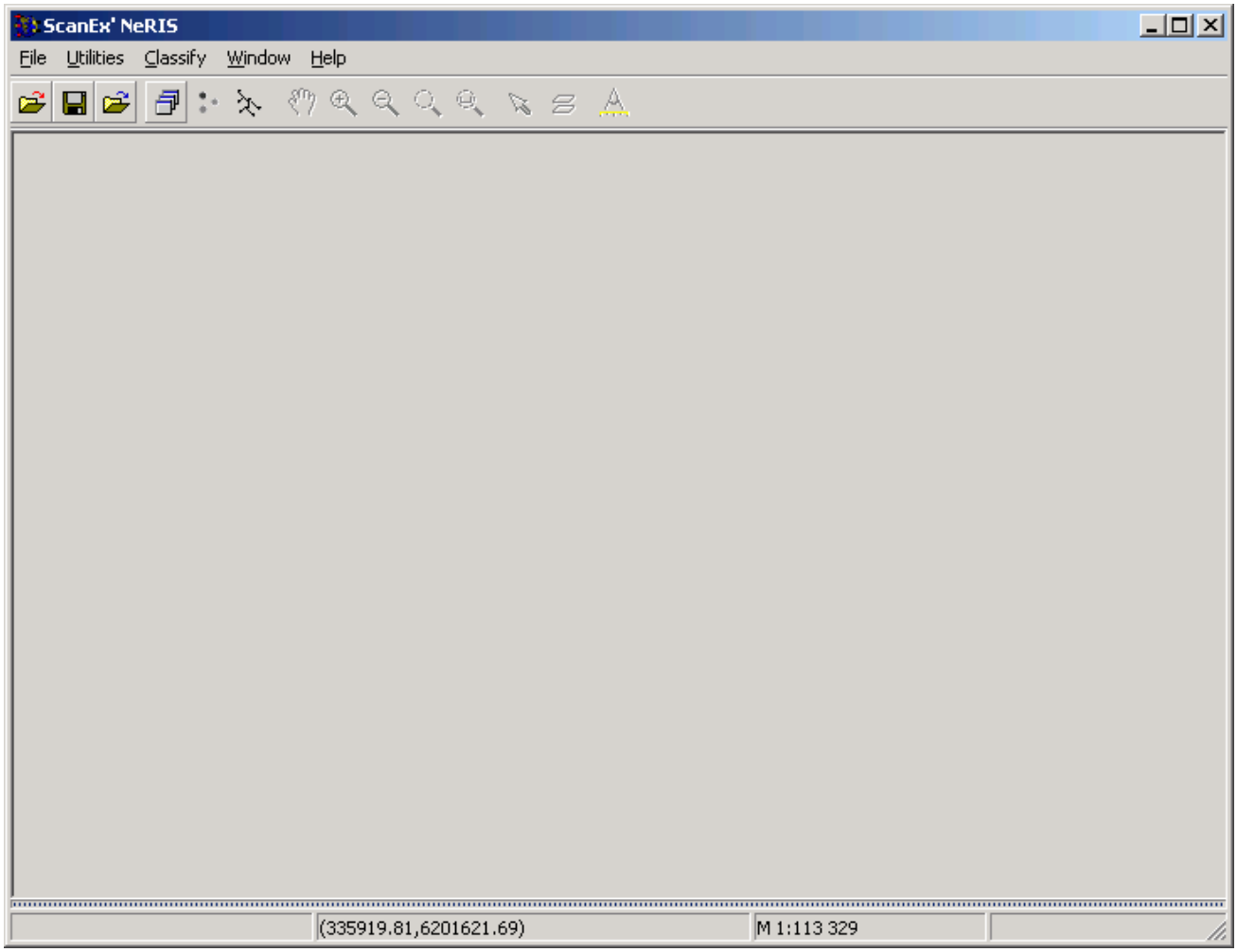
Contextual neural networks

Besides the aforementioned post-processing method using local window, the application also uses contextual neural nets, which apply classes' distribution within the local window as initial data for neural network training. Further use of such neural networks (calibration, classification and thematic classification) is the same as for regular neural nets. The principal difference is in the fact that the raster slot with regular neural net classes is used as initial data.

NeRIS application controls

NeRIS application main dialog

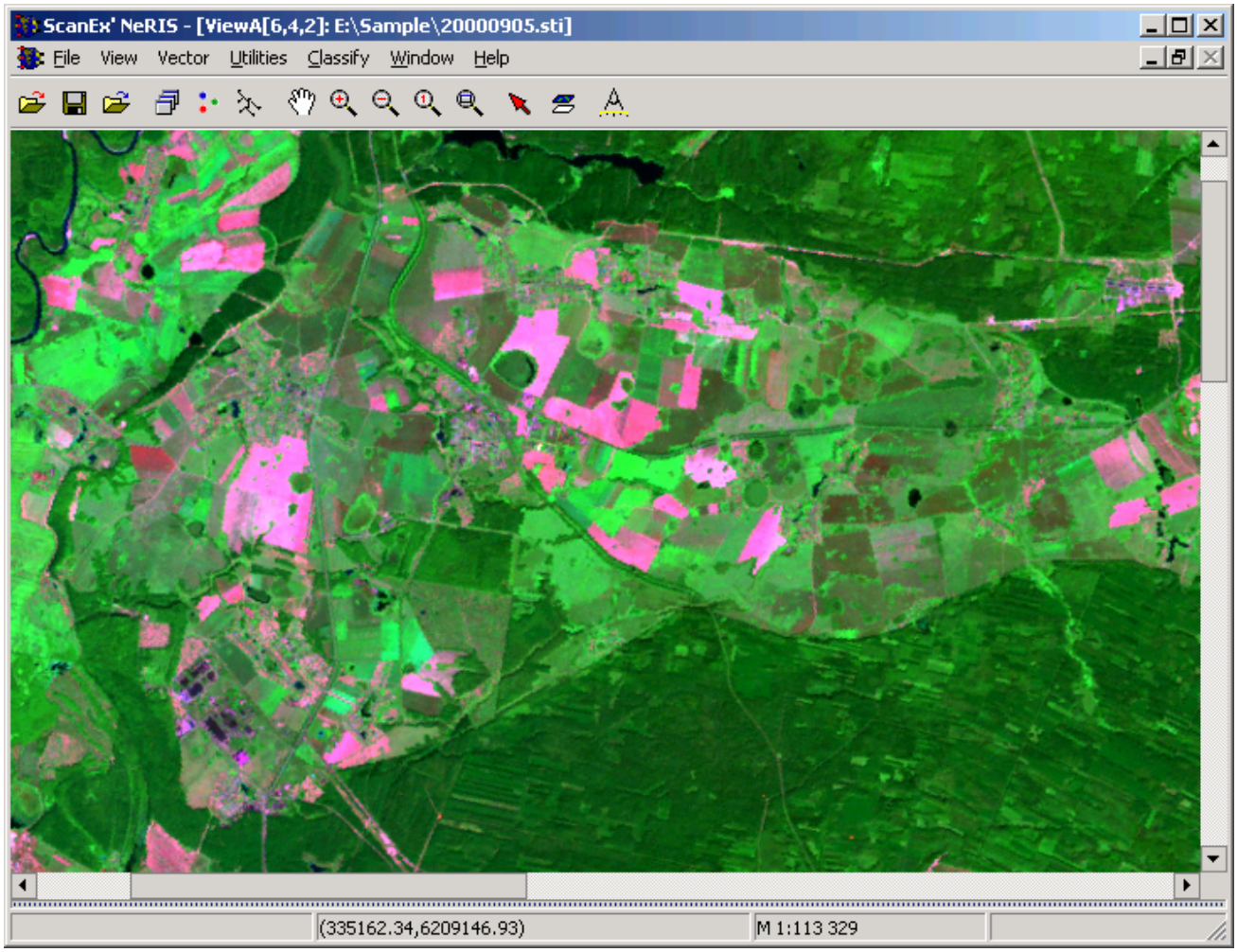
Main **ScanEx' NeRIS** application dialog appears on the screen after loading the program:



As the application interface uses "multi-document" style of the main dialog, all created raster and vector data views are auxiliary to the main dialog and appear only inside it.

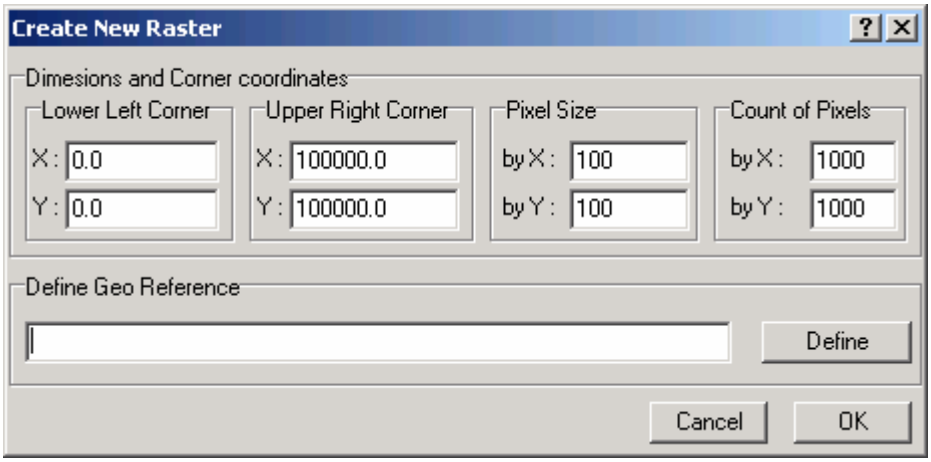
Status bar is shown at the bottom of the main dialog, indicating the current operation mode, mouse coordinates (and the class number of the corresponding pixel during classification results visualization), measured distance (in distance measurement mode) and the current input scale.

If at least one window map is created, the application menu expands (as compared to the one shown above) by adding new controls.


ScanEx' NeRIS - [ViewA[6,4,2]: E:\Sample\20000905.sti]
File View Vector Utilities Classify Window Help
(335162.34,6209146.93)
M 1:113 329

***Creating new raster file in the NeRIS inner format (Create New Raster** dialog)*

The dialog is used to create a blank raster file (containing one clean plane at the time of creation) with specified dimensions and coordinate system.



Lower Left Corner *fields* contain the lower left corner coordinates of the raster being created in the target coordinate system.

Upper Right Corner *fields* contain the upper right corner coordinates of the raster being created in the target coordinate system.

Pixel Size *fields:*

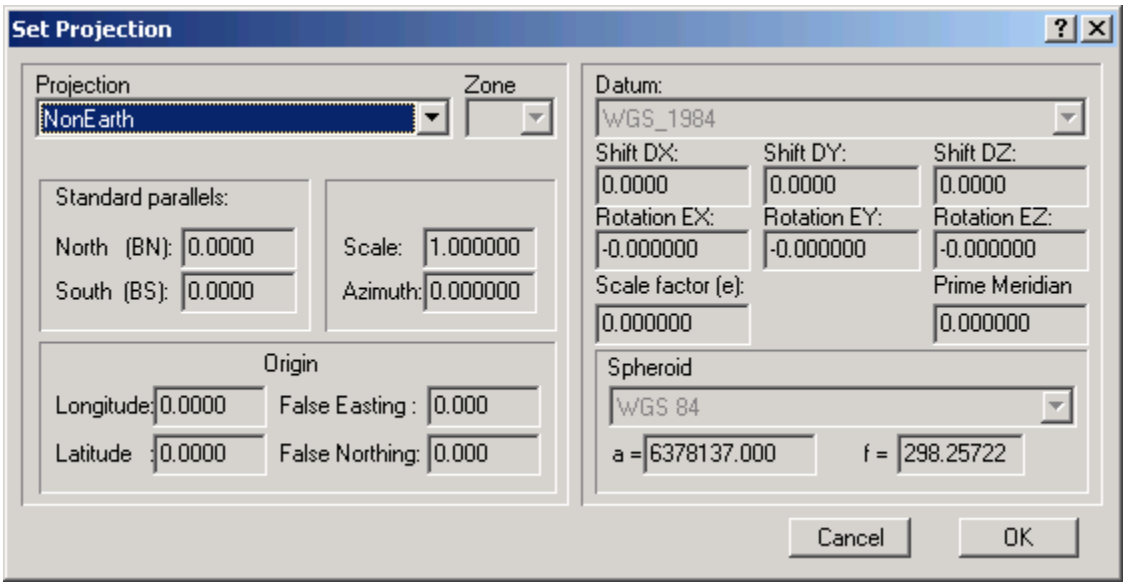
by X contains the horizontal size of the of the created raster (pixels);

by Y contains the vertical size of the of the created raster (pixels).

Count of Pixels *fields***:**

by X contains the horizontal size of the of the created raster (in target coordinate system units);

by Y contains the vertical size of the of the created raster (in target coordinate system units).



Projection *list box* contains the following projection types:

- **"NonEarth"** – "No projection (plan)"
- **"Longitude/Latitude"** – "Longitude/latitude "
- **"Albers Equal Area Conic"** – " Albers Equal Area Conic "
- **"Azimuthal Equidistant"** – " Azimuthally equal distant projection"
- **"Equal Area Cylindrical"** – " Equal Area Cylindrical projection"

- **"Ekert Pseudocylindrical, Number IV"** – " Ekert Pseudocylindrical, Number IV "
- **"Ekert Pseudocylindrical, Number VI"** – " Ekert Pseudocylindrical, Number VI "
- **"Equidistant Conic (Simple Conic)"** – " Equidistant Conic (Simple Conic) projection"
- **"Gall"** – "Gall cylindrical projection "
- **"Rectified Hotine Oblique Mercator"** – " Rectified Hotine Oblique Mercator projection"
- **"Lambert Azimuthal Equal Area"** - " Lambert Azimuthal Equal Area projection"
- **"Lambert Conformal Conic 2SP"** - " Lambert Conformal Conic 2SP projection"
- **"Lambert Conformal Conic 2SP, Belgium"** - " Lambert Conformal Conic 2SP projection (for Belgium)"
- **"Mercator Cylindrical"** – " Mercator Cylindrical projection"
- **"Miller Cylindrical"** – " Miller Cylindrical projection"
- **"Mollweide"** – " Mollweide cylindrical projection"
- **"New Zealand National Grid"** - " New Zealand cartographic grid "
- **"Swiss Oblique Cylindrical"** - " Swiss Oblique Cylindrical projection"
- **"Robinson Cylindrical"** - " Robinson Cylindrical"
- **"Sinusoidal"** - " Sinusoidal projection"
- **"Universal Polar Stereographic"** - " Universal Polar Stereographic projection"
- **"Transverse Mercator"** - " Transverse Mercator projection"
- **"Gauss Kruger"** - " Gauss Kruger projection (Pulkovo 1942)"
- **"Universal Transverse Mercator"** - " Universal Transverse Mercator projection (UTM)"
- **"Cassini"** - " Cassini cylindrical projection"
- **"Polyconic"** - " Polyconic projection"

List box **Zone** enables to select one of the zones (used for Gauss Kruger and UTM projections)

Fields **Standard parallels** (**North (BN), South (BS))** and **Scale** indicate corresponding projection parameters (main (standard) parallels and scale).

Field **Azimuth** enables to set "oblique azimuth ", used in Rectified hotine oblique Mercator projection (set in degrees).

Group **Origin** includes **Longitude**, **Latitude**, **False Easting** and **False Northing** *field*s and sets the starting point of the target coordinate system.

List box **Datum** enables to select parameters for one of the supported reference-ellipsoids:

Shift DX specifies the shift in X-direction;

Shift DY specifies the shift in Y-direction;

Shift DZ specifies the shift in Z-direction;

Rotation EX specifies rotation around X-axis (in seconds);

Rotation EY specifies rotation around Y-axis (in seconds);

Rotation EZ specifies rotation around Y-axis (in seconds);

Scale factor (e) specifies scale correction coefficient (in parts per million);

Prime Meridian specifies prime meridian longitude (in degrees east of Greenwich).

List box **Spheroid** enables to select one of the supported spheroids.

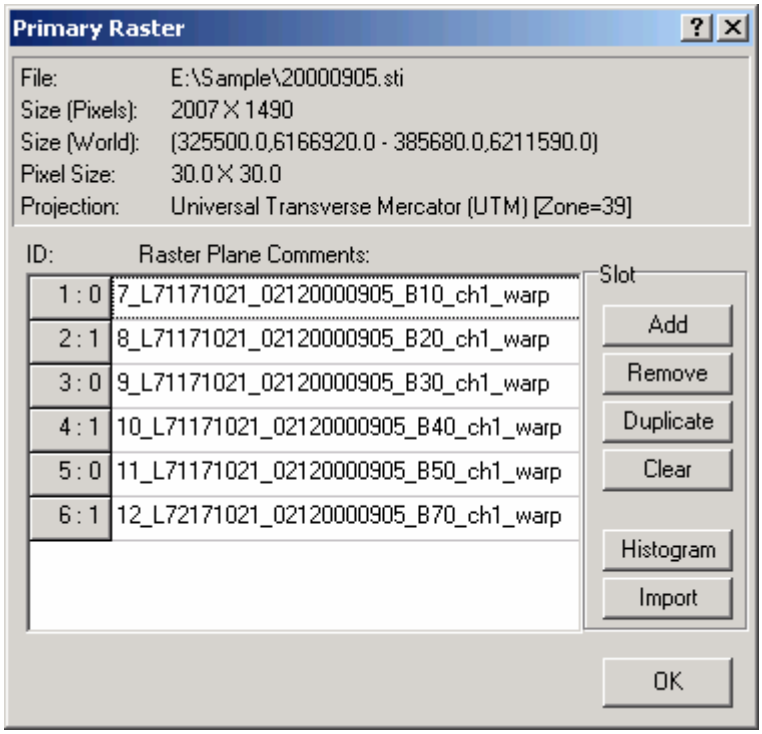
Field **a –** specifies semi-major axis of the spheroid (in meters).

Field **f –** specifies flattening factor.

Created raster can be saved to a file according to standard procedure (using **File/Save Primary Raster** menu) or using relevant hot key button.

***Manipulations with primary and secondary raster (Primary Raster** dialog)*

After selecting raster controls from **Utilities** menu (**Primary Raster, Secondary Raster** items), the controls appear on the screen as shown below:



Information fields, displaying sizes and raster projection are in the top of the dialog.

Primary Raster *dialog* contains two columns for each raster slot used:

- *Column* **ID** specifies the internal slot identification and (after colons) the number of views, where this slot was used (the use of slot during the creation of views inhibits all its updating activities).
- *Column* **Raster Plane Comments** contains this slot description (description can be edited using standard Windows tools by making this field active using the mouse).

Selected slot in the list is the current one; all operations are applied only to the current slot.

Button **Add** enables to create a blank slot (filled with zero values) and to add it to raster.

Button **Remove** enables to delete the current slot from the raster.

Button **Duplicate** enables to create a copy of the current slot and to add it to raster.

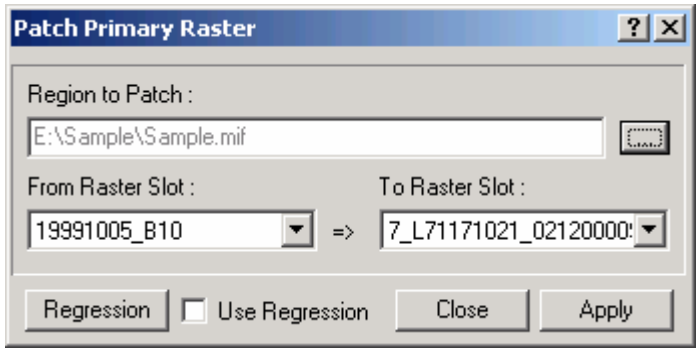
Button **Clear** enables to clear the current slot (fill it with default value, normally zero).

Button **Histogram** enables to renumber histogram of the current slot.

Button **Import** enables to import raster data into PPM, PGM, Surfer GRD, ArcInfo ASCII, BMP formats and to save it as slot (slots).

***Creating mosaics from several images (Patch Primary Raster** dialog)*

To control mosaics creation, use **Patch Primary Raster** item in **Utilities** menu. The dialog with controls, indicated below, appears on the screen:



*"***Region to Patch***"* *push-button* is used to outline the region borders (in the target coordinate system of the primary raster), which should be replaced by the relevant region from the indicated slot of the secondary raster.

List box **From Raster Slot** specifies the secondary raster slot, where the "patch" is taken from.

List box **To Raster Slot** specifies the primary raster slot, where the "patch" is put to.

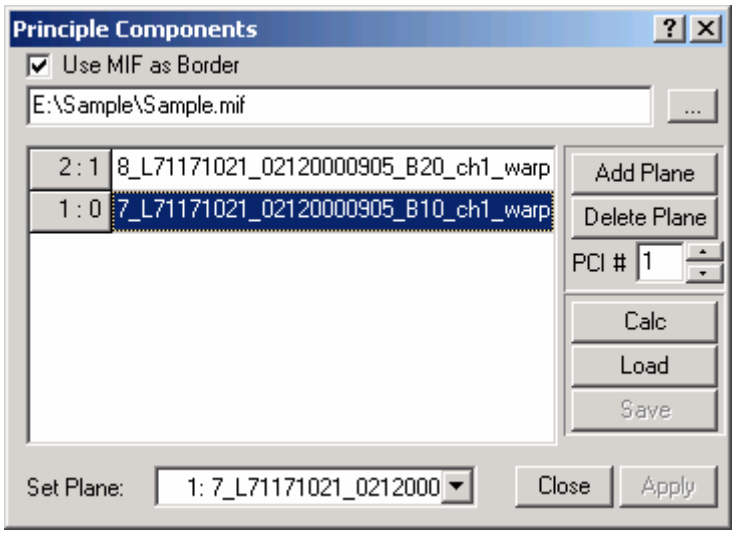
Button **Apply** initiates the operation.

Button **Regression** calculates the linear regression to equalize the "patch" and image brightness histograms.

Checkbox **Use Regression**. If checked, the previously built regression is used to modify the "patch" brightness. Regression can be built using one file with borders (where slots images are partially overlapped), whereas the "patch" itself will have different borders.

***Principle Components calculation (Principle Components** dialog)*

To calculate principal components of several primary raster slots (eigenvectors of the corresponding covariance matrix or the values of principal components for each raster point), use **Principle Components** item in **Utilities** menu. Dialog with the controls, indicated below, appears on the screen.



Checkbox **Use MIF as Border** enables to specify the requirement to set raster borders, where the covariance matrix eigenvectors or principal components are calculated (the entire raster is calculated by default).

Region border file selection push-button is used to outline the region (in the target coordinate system), where the eigenvectors and principal components are calculated.

Table of raster planes contains two columns for each primary raster slot used in calculation:

- Left **ID c**olumn of the table displays the internal slot identification and (after colons) the number of views, where this slot was used (the use of slot during the creation of views inhibits all its updating activities).
- Right **Raster Planes** column of the table contains this slot description.

List box **Set Plane** changes the current slot to the specified primary raster slot.

Calculating covariance matrix and eigenvectors

Button **Add Plane** adds one slot to the list for covariance matrix and its eigenvectors calculation.

Button **Delete Plane** deletes the current slot from the list for covariance matrix and its eigenvectors calculation.

Field **PCI#** contains the number of eigenvectors being calculated (corresponding to the specified number of maximum eigenvalues of covariance matrix).

Button **Calc** initiates the calculation of covariance matrix and its eigenvectors.

Button **Save** enables to save the calculated covariance matrix and its eigenvectors to a file on the disk.

Button **Load** enables to load previously calculated covariance matrix and its eigenvectors from a file on the disk.

Principal components calculation and saving

Button **Apply** initiates the calculation of the specified number of raster planes with principal components and saving them in new slots, added to primary raster. As only 8 bits are used to save numbers in slots, the calculations are made in two steps. At first, maximum and minimum values are calculated, which the principal component acquires, then these values are used for the linear transformation to the 0-255 range of values (0 – is the minimum principal component value, 255 – maximum) and saving of received specified values. The principal component dispersion data is stored in the slot description.

***Using filters for image changes (Filter Plane** dialog)*

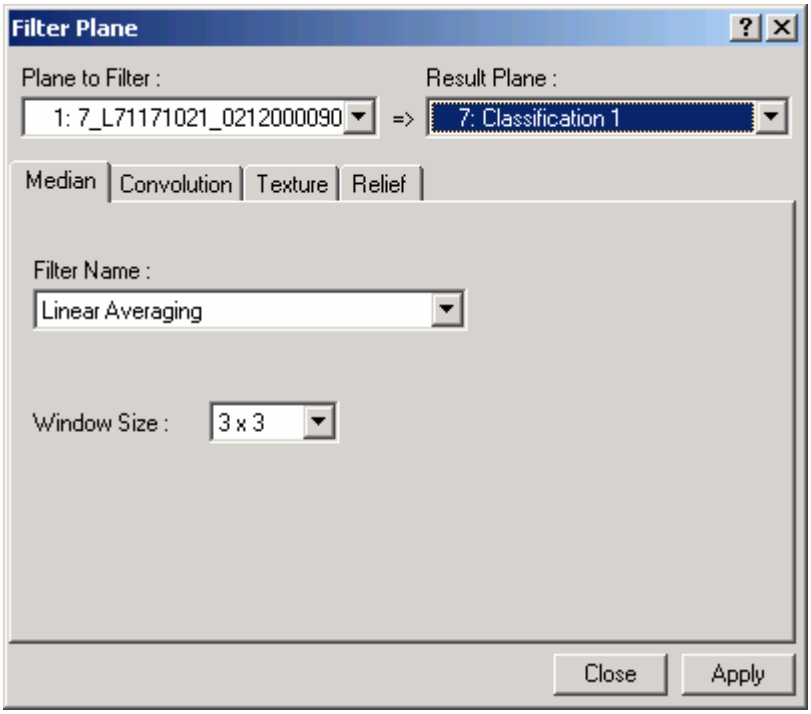
To apply filters to the primary raster slots, use **Filtration** item in **Utilities** menu. Filtration control dialog, indicated below, appears on the screen:

List box **Plane to Filter –** selects the primary raster slot, which the filter will be applied to.

List box **Result Plane** – selects slot for the results.

Filtration *tabs*– a set of fields to select filter type and its parameters. Currently, the following filter types are available: statistic (**Median** *tab*), convolution filters (**Convolution** *tab*), calculation of image texture features in local window (**Texture** *tab*) and "brightness" classification (**Relief** *tab*).

***Using statistic filters (Median** tab)*

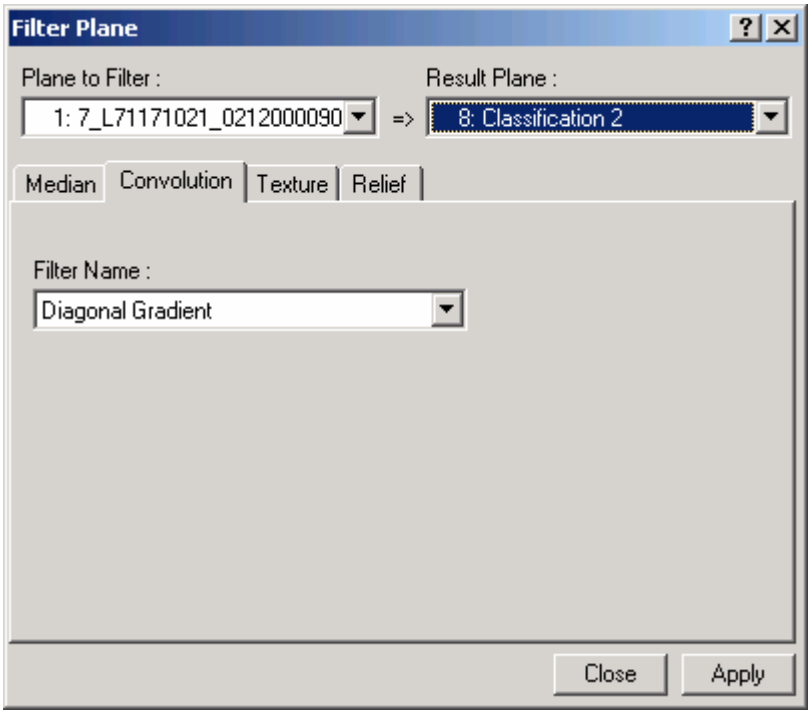


List box **Window Size** specifies the size of local window, used during filtration.

List box **Filter Name** specifies the type of the used filter:

Linear Averaging	Use average value in window
Median Filter	Use value median in window
Majority Filter	Use the value, most encountered in window

***Using convolution filters (Convolution** tab)*



List box **Filter Name** specifies the type of the used filter:

Diagonal Gradient	3x3 filter to separate diagonal gradients (angle 45°)
Back Diagonal Gradient	3x3 filter to separate diagonal gradients (угол 135°)
Laplacian	3x3 Laplacian transformation filter
Circular Marr (Laplacian-of-Gaussian)	11x11 Marr filter (Gaussian and Laplacian transformation superposition)
Sharpening (Laplacian+original)	3x3 filter to separate details (Laplacian and original image sum)
Vertical edge (Sobel)	3x3 Sobel filter to separate vertical lines
Horizontal edge (Sobel)	3x3 Sobel filter to separate horizontal lines

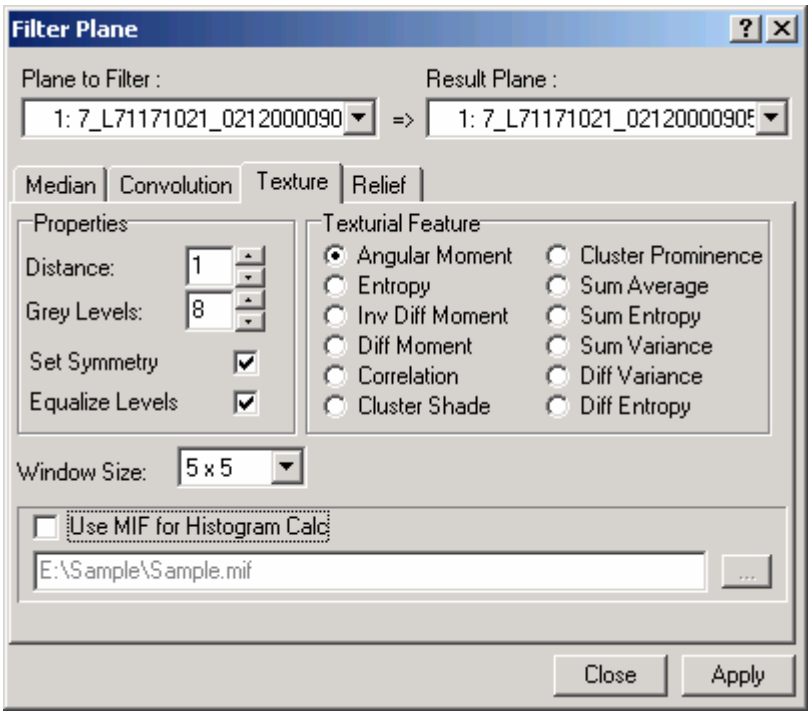
***Calculating texture features in local window (Texture** tab)*

Texture features are calculated based on the analysis of the image brightness level co-occurrence in the local rectangular window, recorded as Grey Level Co-occurrence Matrix (GLCM), first suggested in the Haralik (Haralik et al., 1973) study. Co-occurrence is defined based on a pair of pixels, separated by a given direction and distance. Thus for example, if the distance equals one pixel and direction is horizontal, all pairs of pixels in the window, located next to each other horizontally, are taken for co-occurrence analysis.

As almost all the cells of the built matrix will be blank if the full range of possible brightness levels (256) is used and the window is small, the entire brightness values range of the selected image is divided into intervals, inside which all brightness values are considered similar, to build a **GLC**-matrix. Intervals can have even brightness and number of pixels, found in each interval. The first method can be used if the image brightness histogram does not have evident peaks and covers the whole range of possible values. If the brightness histogram has, for example, only one narrow peak, this method will bring to the creation of a big number of blank matrix cells. The second method is used in this case, based on the calculation of such a set of intervals using the image brightness histogram that the histogram area over each of them is approximately the same (this will result in a better **GLC**-matrix filling rate). Moreover, using the brightness histogram of only a required part of the image, it is possible to improve the GLC-matrix filling and consequently, the quality of the image texture features calculation.

Own GLC-matrix is created for each direction and distance, where the probability of the separated pixels having **i** and **j** brightness is recorded into the cell (**i, j**). Therefore, **GLC**-matrix is square, with vertical and horizontal dimensions, equal to the number of used brightness values (intervals).

In texture classification the matrix derived features are normally used, rather than the **GLC**-matrices themselves.



Checkbox **Use MIF for Histogram Calc** – if unchecked, the entire image is used for histogram calculation, if checked – only the region, specified in the vector file, is used for histogram calculation.

Region border file selection push-button is used to outline the region (in the target coordinate system of the primary raster), where the histogram is calculated.

List box **Distance** specifies the distance, used in **GLC**-matrix calculation (the bigger is the distance, the bigger is the size of the used local window to get the sufficient number of pairs of pixels for statistic calculation).

List box **Grey levels** specifies the number of image intensity values intervals, used in **GLC**-matrix calculations.

List box **Window Size** specifies the local window size, used in **GLC**-matrix calculations.

Checkbox **Set Symmetry** – if unchecked, the obtained **GLC**-matrix is symmetrized.

Checkbox **Equalize Levels** – if unchecked, the brightness intervals are built equal in sizes, if checked –approximately equal in number of pixels inside.

Group of radio buttons **Texturial Features** specifies the texture features being calculated:

Specifying the probability, recorded in cell (**i**, **j**) of **GLC**-matrix through $\hat{f}(i,j)$, we define

$\mu_y = \sum_i i \sum_j \hat{f}(i,j)$	- average in rows
$\mu_x = \sum_j j \sum_i \hat{f}(i,j)$	- average in columns
$\sigma_y = \sum_i (i-\mu_i)^2 \sum_j \hat{f}(i,j)$	- average dispersions in rows
$\sigma_x = \sum_j (j-\mu_j)^2 \sum_i \hat{f}(i,j)$	- average dispersions in columns

and the texture features:

Angular Moment	**ANGULAR MOMENT,** $\sum_{i,j} \hat{f}(i,j)^2$

Entropy	entropy, $-\sum_{i,j} \hat{f}(i,j) \log \hat{f}(i,j)$
Inv Diff Moment	Inverse difference moment, $\sum_{i,j} \frac{1}{1+(i-j)^2} \hat{f}(i,j)$
Diff Moment	Difference moment, $\sum_{i,j} (i-j)^2 \hat{f}(i,j)$
Correlation	correlation, $\frac{\sum_{i,j} \left((i-\mu_y) \cdot (j-\mu_x) \right) \hat{f}(i,j)}{\sigma_y \cdot \sigma_x}$
Cluster Shade	cluster "shade", $\sum_{i,j} \left((i-\mu_y) + (j-\mu_x) \right)^3 \hat{f}(i,j)$
Cluster Prominence	Cluster "prominence", $\sum_{i,j} \left((i-\mu_y) + (j-\mu_x) \right)^4 \hat{f}(i,j)$
Sum Average	Sum average, $\sum_k k \sum_{i+j=k} \hat{f}(i,j)$
Sum Entropy	- sum entropy, $-\sum_k \sum_{i+j=k} \hat{f}(i,j) \cdot \log \left(\sum_{i+j=k} \hat{f}(i,j) \right)$
Sum Variance	Sum variance, $\sum_k (k-\mu_s)^2 \sum_{i+j=k} \hat{f}(i,j)$, where μ_s - sum average
Diff Variance	Difference variance, $\sum_k (k-\mu_d)^2 \sum_{i-j=k} \hat{f}(i,j)$, where μ_d - sum average
Diff Entropy	Difference entropy, $-\sum_k \sum_{i-j=k} \hat{f}(i,j) \cdot \log \left(\sum_{i-j=k} \hat{f}(i,j) \right)$

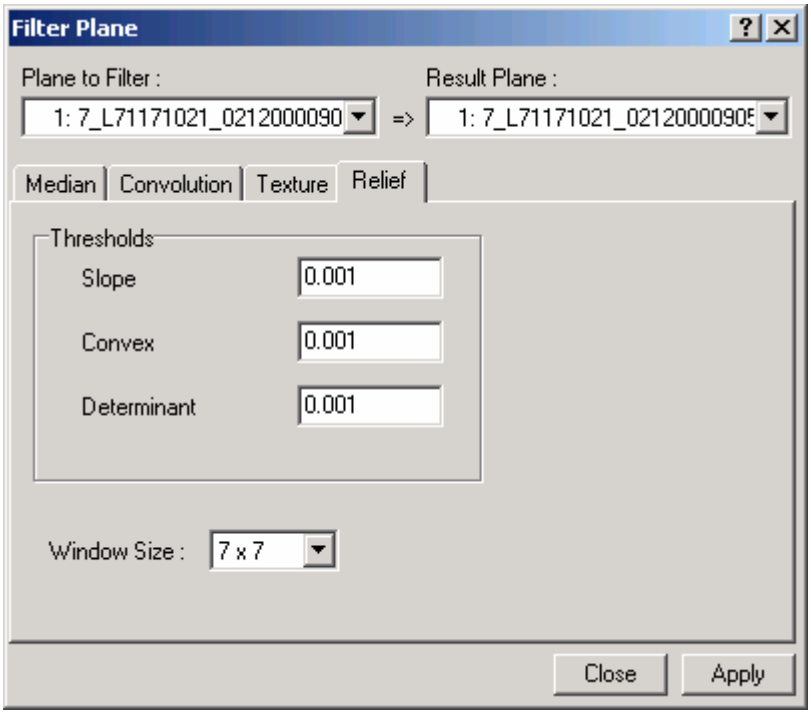
Classification of image "brightness relief" (Relief** tab**)

This filter takes brightness values as relief vertices and classifies each point based on local quadratic model, built using local window.

Below are the codes, recorded by the program as a result (processing results should be displayed using legend; the biggest part of the results is usually taken by codes 1 and 2):

1	Watershed
2	Stream flow (river bed)
3	Flat place (plain)
4	Peaks (vertices)
5	Col (selliform)
6	Depression (water basin bed)

It is possible to control the scale of classified morphological objects by changing the window size.



List box **Window Size** specifies the window size to build the model.

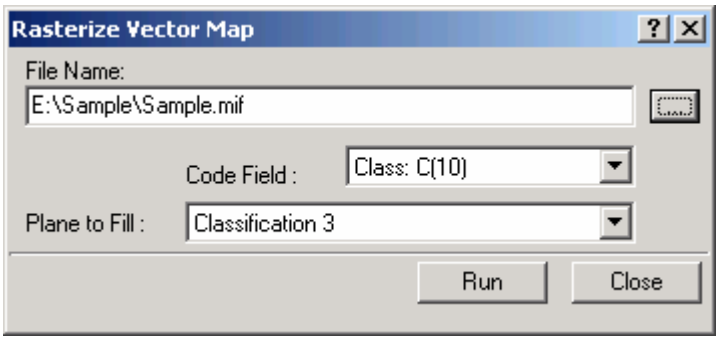
Field **Slope** specifies slope threshold value (in radians), below which the parameter is considered as zero.

Field **Convex** specifies surface curvature threshold value (in meters) below which the parameter is considered as zero.

Field **Determinant** specifies internal thresholds to consider slope (in radians), surface curvature (in meters) and model determinant (quadratic form) values as zero.

***Moving data from vector maps into raster slots (Rasterize Vector Map** dialog)*

To move data from vector maps into the primary raster slots, use **Rasterize Vector Map** item in **Utilities** menu. Rasterizing control dialog appear on the screen.



Map file name selection button. The file name is used to indicate the file, containing borders of the region to be vectorized.

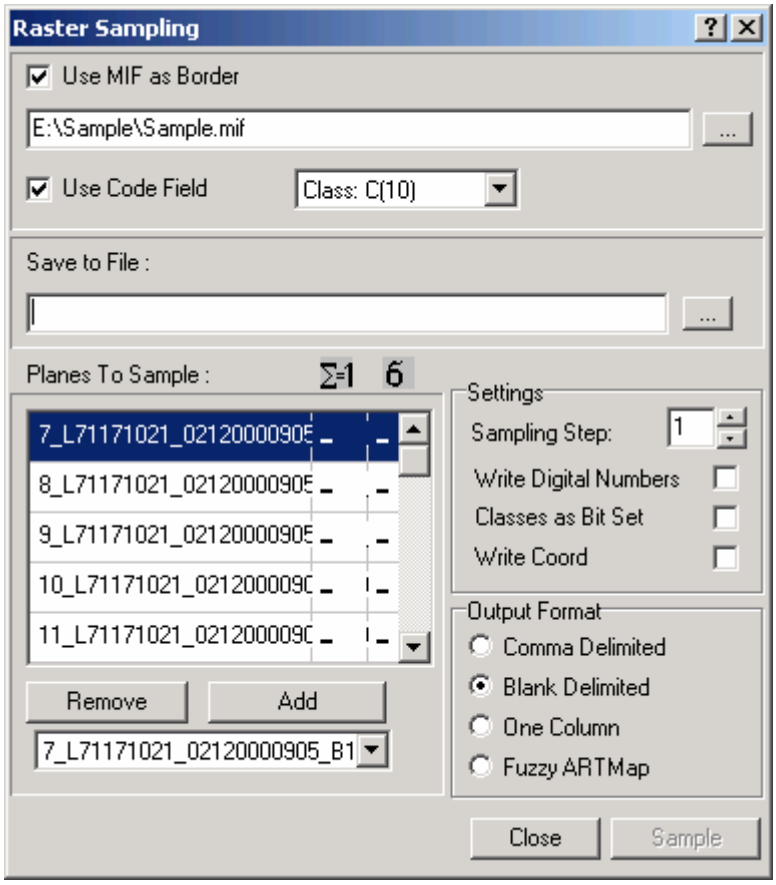
List box **Code Field** enables to select the vector map attributive table field, containing the value recorded for each region into the final slot.

List box **Plane to Fill** enables to select the primary raster slot, where rasterizing results will be recorded.

Button **Run** initiates rasterizing.

***Creating sampling text file from primary raster (Raster Sampling** dialog)*

To move data from primary raster slots into text files, use **Raster Sampling** item in **Utilities** menu. Controls dialog appears on the screen as shown below.



Checkbox **Use MIF as Border** – if checked, the region outlined by the specified vector map is used for sampling (if not checked, sampling will be done for the entire raster).

Border file selection button– defines the vector map name, containing one or several regions, outlining the region used for sampling.

Checkbox **Use Code Field** – if checked, data from the specified vector map attributive table field is moved into the file in form of a separate column for sampling.

List box **Use Code Field specifies** the vector map attributive table field, where the value for sampling is taken from.

File selection button **Save to File** *specifies* the text file name, where the sampling will be recorded to.

Table **Planes to Sample** contains the list of primary raster slots, involved in sampling. There are two columns for each of them:

- Primary raster slot description column.
- Column indicating the requirement to normalize created file rows (vector length, made of the pixel values in the slots marked with + will equal to one).

Button **Remove** removes the current slot from the list for sampling.

Button **Add** adds a new slot into the list for sampling.

List box **Planes** below changes the current slot in the sampling list for one of the primary raster slots.

Field **Sampling Step** specifies the step for raster pixel selection for sampling (if 1 – all pixels are selected, if 2 – every second in the row, and so on).

Checkbox **Write Digital Numbers (0-255) –** if checked, the output values are scaled within 0-255 range, otherwise values from 0 to 1 with the step of 1/255 are displayed.

Checkbox **Classes as Bit Set –** if checked, for each recorded sampling row the region code from the vector map attributive table is presented as a sequence of zeros and units, containing one unit on the place, corresponding to integer

code (if code value equals to 2 and the maximum code value in the file equals to 5, the sequence 0,1,0,0,0 is produced in the sampling row on the place of the code).

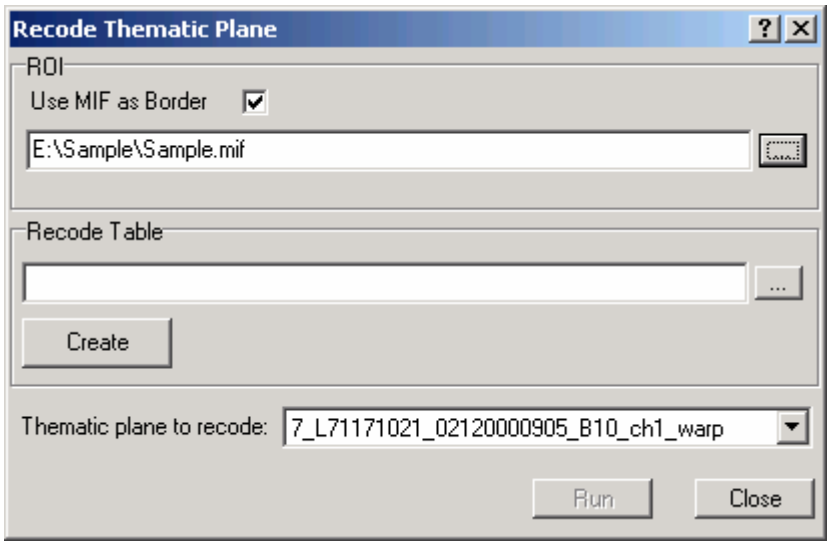
Checkbox **Write Coord** – if checked, center coordinates of the relevant pixel are added to each output file row.

Radio button group **Output Format** specifies the output file format:

Comma Delimited	In rows, delimited by commas
Blank Delimited	In rows, delimited by blanks
One Column	In one column
Fuzzy ARTMap	In **ARTGallery** program format

Button **Sample** – initiates sampling. Transformation of values, taken from the relevant slots, is the same (normalization) as during neural networks training.

***Creating recoding tables (Recode Thematic Plane** dialog)*



Checkbox **Use MIF as Border** specifies the requirement to limit working area with regions, contained within vector map file.

Border file selection button enables to select vector map file, containing working area borders.

Field **Recode Table** enables to select the name of table, containing recode values.

Button **Create** opens dialog, enabling to create recode table.

List box **Thematic plane to recode** specifies primary raster plane, which needs to be recoded.

Button **Run** initiates recoding.

***Layers controls (Layers control** dialog)*

To control vector layers, use **Layers Control** item in **Vector** menu of the main display window menu. Vector layers control dialog appears on the screen.

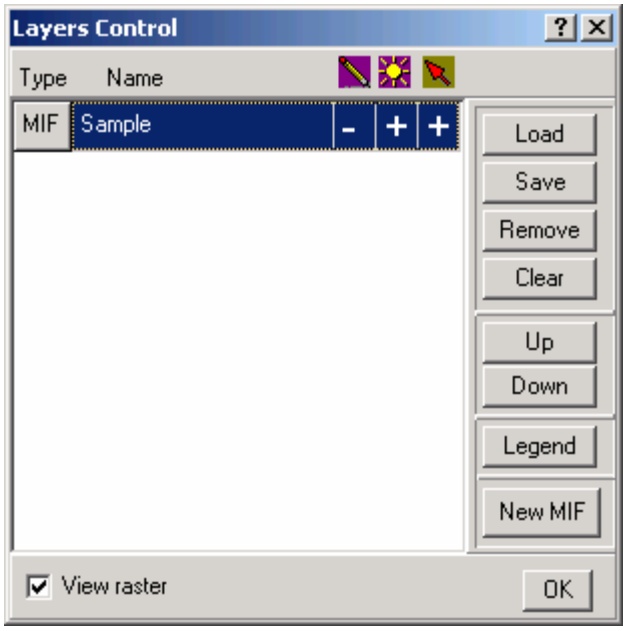


Table **Vector Layers Control** contains the list of loaded vector layers with information and controls for each. There are five columns available for each layer:

- Column with the loaded vector layer type (MIF in this version –MapInfo program exchange format).
- Column with the layer name (file name, without extension or path, where this layer is loaded from).
- Column with the layer editing feature
- Column with the layer visibility feature
- Column with the layer objects selection possibility feature

Note: *All layer features are changed by clicking left mouse button on the corresponding cell.*

Button **Load** loads vector layers from file. After loading, the layer will be stored in RAM, so one layer can be loaded several times and changed differently, then saved into different files.

Button **Save** saves current vector layer to file.

Button **Remove** removes current vector layer from RAM.

Button **Clear** clears all objects of current layer.

Buttons **Up/Down** enable to change vector layers drawing sequence and their preview when searching for objects.

Button **Legend** enables to control thematic vector layers display (see corresponding dialog description).

Button **New MIF** creates a new vector layer.

Checkbox **View raster** enables to deactivate raster drawing in the display window (in this case only vector layers are drawn in all windows).

***Vector layers thematic display control (Legend** dialog)*

To control vector layers thematic display, use **Legend** dialog, which can be opened in vector layers control dialog.

It enables to select one of the three methods of object style display:

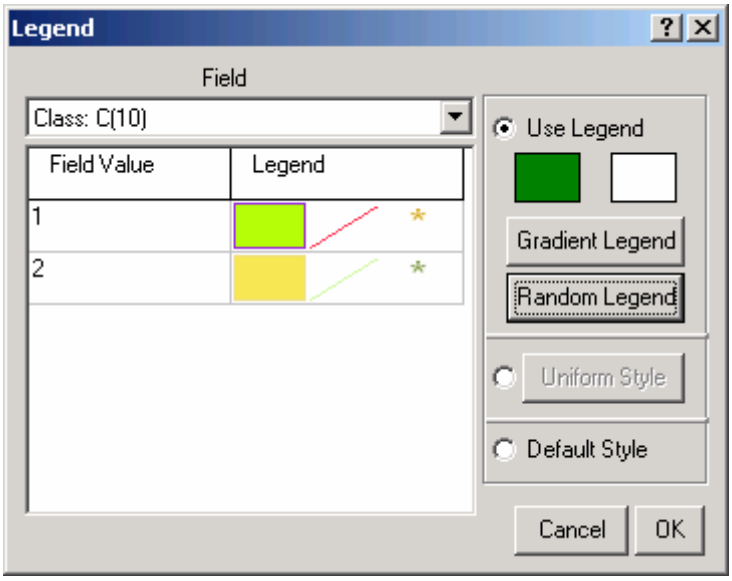
- Using values of one of the fields to specify style (thematic legend).
- Using one style to display all layer objects.
- Using the style, which is saved for each object in the vector layer.

Objects output style determination method:

- **Legend** – create thematic legend in accordance with the attribute values.
- **Determine** – display all objects using one style.
- **By default** – use default objects style.

As the controls depend on the selected display method, we will describe thematic legend creation controls first.

Display according to thematic legend



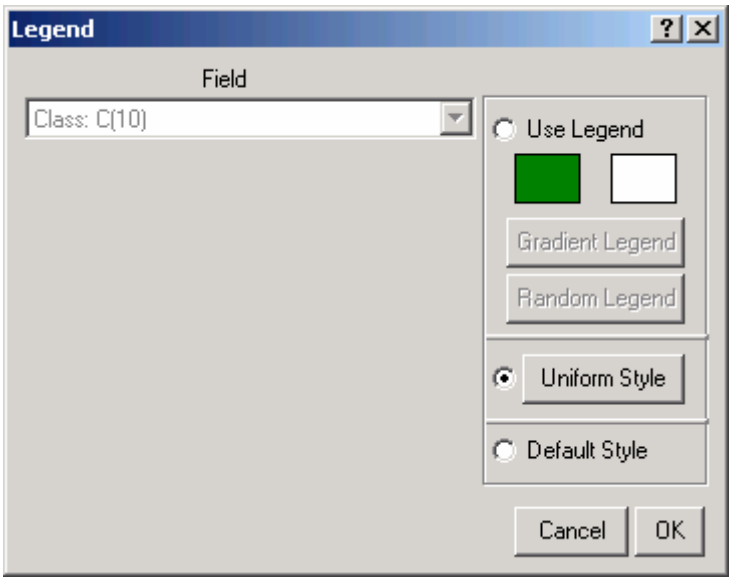
List box **Field** provides for the selection of vector layer attributive table field, according to which a legend will be created. In this case the simplest version is used, when the number of different field values should not exceed 256.

Table **Field Value/Legend** contains the list of the selected field values (one per row), and for each of them – display style for point, linear and areal objects. To change the style, click left mouse button on the style cell to open vector objects style change dialog.

Button **Gradient Legend** enables to obtain gradient color in objects display styles (initial and final colors are selected by clicking left mouse button on the corresponding rectangular).

Button **Random Legend** enables generate display style at random.

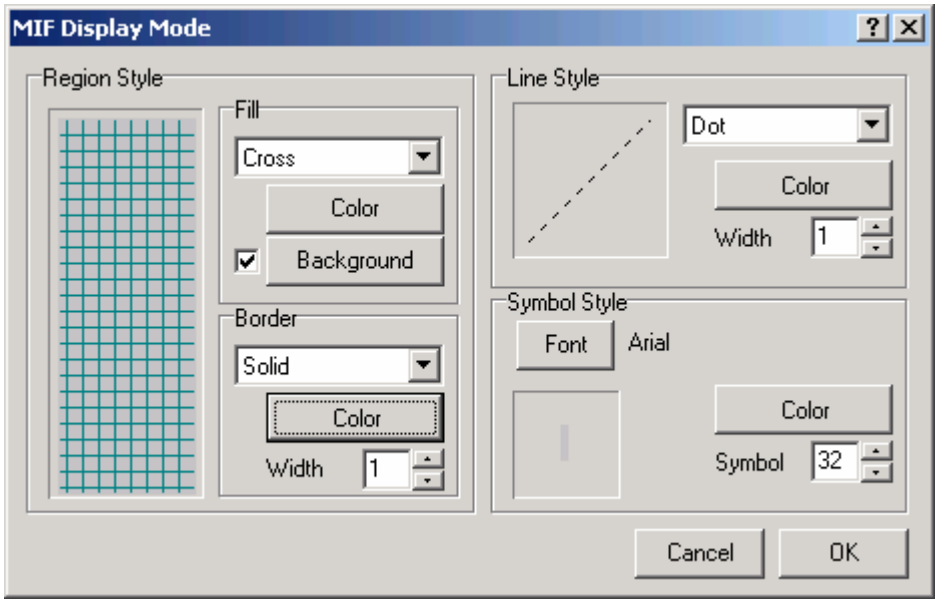
Display of all objects with one style



Button **Uniform Style** opens the vector objects display style selection dialog, which enables to select the uniform style of point, linear and areal objects display.

***Vector objects display mode control (MIF Display Mode** dialog)*

This dialog enables to control vector objects display mode.



This dialog contains three groups of controls for the point, linear and areal objects respectively.

Group **Region Style** *of the areal objects display style control*

List box **Fill** enables to select one of the eight standard MS Windows filling and lining styles.

Button **Color** enables to select (using standard MS Windows dialog for color selection) the filling/lining color.

Checkbox **Background** provides for background fill control (if unchecked, background is not filled).

Button **Background** enables to select lining background (using standard MS Windows dialog for color selection).

List box **Border** enables to select of the six standard MS Windows line styles for drawing borders.

Button **Color** under enables to select border color (using standard MS Windows dialog for color selection).

Field **Width** specifies border drawing line thickness.

Rectangular display preview using the selected style is available in the corresponding field for control.

Group **Line Style** *of linear objects display style control*

List box **Line style** enables to select one of the six standard MS Windows style for line drawing.

Button **Color** enables to select the line color (using standard MS Windows dialog for color selection).

Field **Width** specifies line thickness when drawing.

Line display preview using the selected style is available in the corresponding field for control.

Group **Symbol Style** *of point objects display style control*

Button **Font** controls font selection for point objects drawing.

Button **Color** enables to select the symbol color (using standard MS Windows dialog for color selection).

Field **Symbol** specifies the displayed symbol number.

Font name and selected symbol are displayed in the corresponding field for control.

***Vector object data modification (Edit Object Data** dialog)*

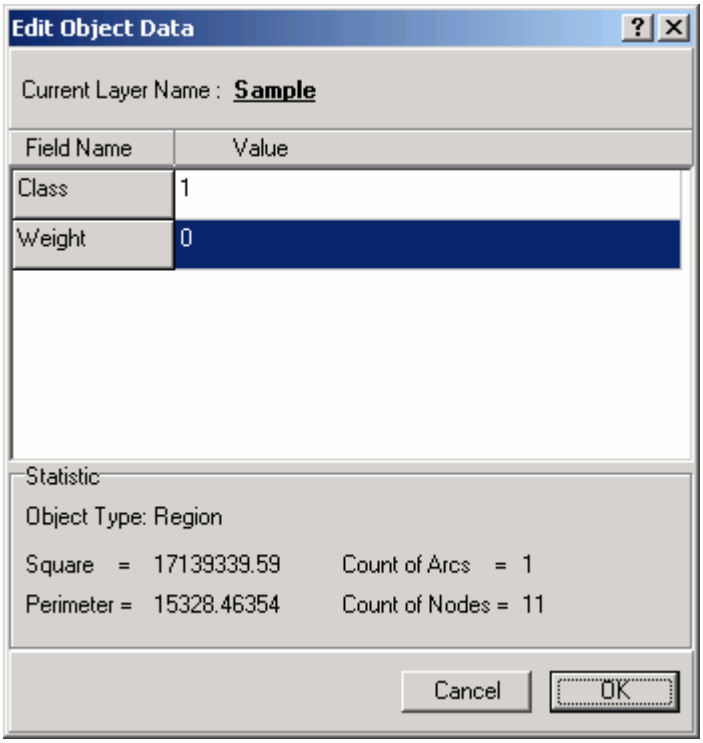


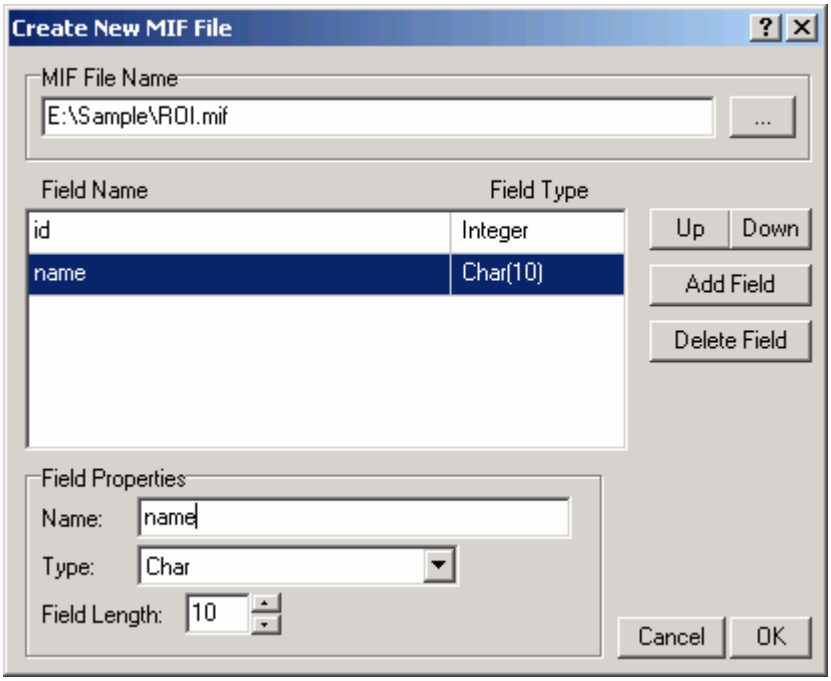
Table **Field name/Value** contains the list of the vector layer attributive table fields. There are two columns available for each of them:

- Column with the field name
- Editable column with the field value

Statistic data is available at the bottom about the selected object (object type, area, perimeter, number of arcs and nodes).

***New vector layers creation control (Create New MIF File** dialog)*

To create new vector layers (**MIF/MID** format files in the current software version), use the following dialog:



File Name selection button to record the new vector layer is used to select the file type and name (only **MIF/MID** format files creation is supported by the current software version).

Table contains the list of vector layer attributive table fields. There are two columns available for each of them:

- Column **Field Name**.
- Column **Field Type**.

Selection of the current field in the table enables to edit its properties using the following controls.

Field **Name** specifies and modifies the field name.

List box **Type** enables to select one of the available field types (**Char, Integer, Short, Float, Decimal, Date, Logical**) for **MIF/MID** files.

Field **Field Length** specifies the field length (where length has no evident result from the field type).

Buttons **Up** and **Down** enables to change field sequence in table by moving the current field up or down.

Button **Add Field** adds new fields in table.

Button **Delete Field** deletes current field from table.

***Raster display control (View parameters** dialog)*

One of the basic operations, used by **NeRIS** application, is making primary and secondary raster display. Both source images and derived images after processing and classification can be used as initial data. Another important operation using the controls of this dialog is to transform classification results into vectors and to store them in one of the loaded vector layers.

The application allows for several display methods:

- **RGB** composite image, using three primary or secondary slots.
- Black-and-white composite (grey-scale) image, using one primary or secondary raster slot.
- Image generation using the number of colors equal to the number of classes, contained in one slot of the primary or secondary raster, received after classification.

Display mode is specified using **Select Raster, View Type** and **Type** controls of **View Parameters** *dialog form***.**

View Type *group of radio-button* specifies the number of slots used to make the display.

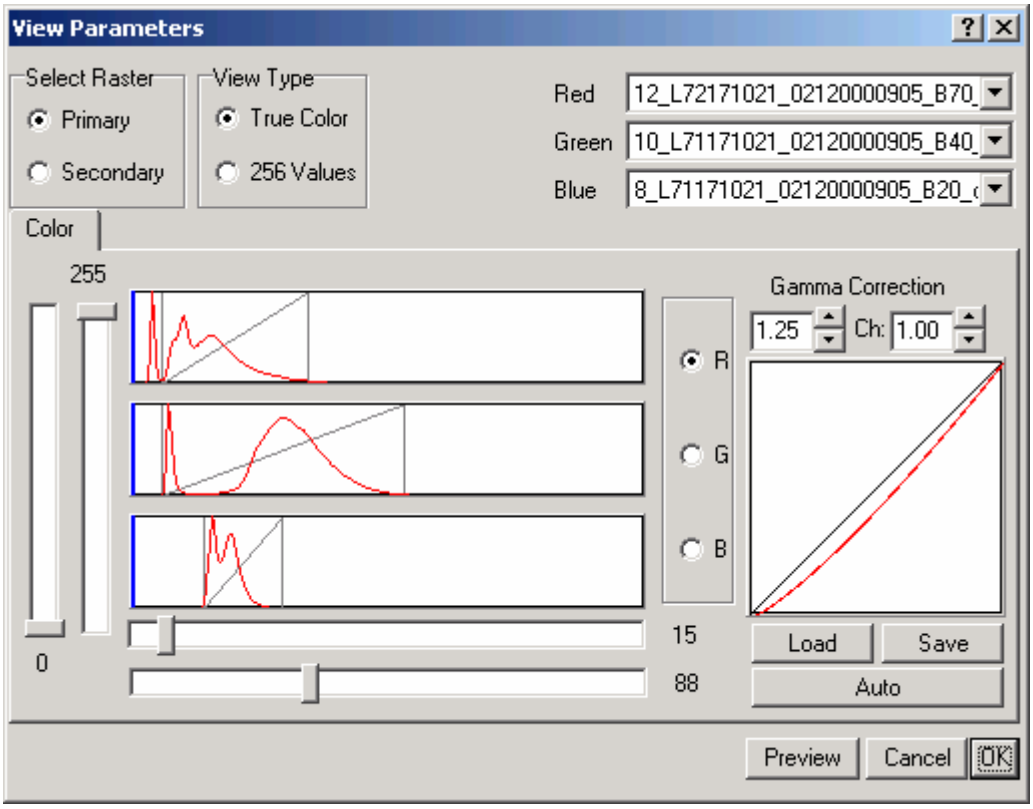
- **3 channels** value **(True Color)** means that the display is formed of three slots, which values are interpreted accordingly as red (**Red**), green (**Green**) and blue (**Blue**) color tones of the created images, whereas the derived colors are defined by the linear transformations and gamma-correction of these values.
- **1 channel** value **(256 values)** means that the display is formed of one slot, which values are interpreted either as brightness or as the class number, depending on the **Type** control.

Type *group of radio-buttons* specifies the slot value interpretation method when making a display.

- **SOM** selection means that the values are interpreted as numbers of classes, received as a result of neural net classification. In this case the colors for each class display are set via the corresponding dialog control.
- **Grey** selection means that the values are interpreted as brightness and the grey levels for each value are defined by linear transformation and gamma-correction of these values.
- **Legend** selection means that the values are interpreted as numbers of classes, received as a result of neural net classification or image filtration. In this case the colors for each class display are set via the corresponding dialog control.

Making RGB view

Dialog used to make **RGB** display is shown below:



Select Raster *group of radio-buttons* controls the selection of primary (**Primary)** or secondary (**Secondary**) raster as the source of slots for the view.

List boxes **Red, Green** and **Blue** specify slots for the red, green and blue planes of the view being created respectively.

There are three rectangular in the middle of the dialog to select the current slot, as well as to output the data about the selected slots: the histogram (shown in red), maximum and minimum output values (shown in blue to the left), upper and lower thresholds of the input image (shown in grey vertical lines) and gamma-correction curve dialog, also shown in grey. Current active slot is reflected in the option group to the right.

RGB *group of radio-buttons* enables to select the current slot for its display control.

Horizontal slider bars. Two horizontal controls below the histogram provide for setting up upper and lower input values thresholds, used during the transformation of current slot values (values below the threshold are shown as 0, values over the threshold are shown as 255, other values have linear transformation in the interval within the specified values).

Vertical slider bars. Two vertical controls to the left of the histogram provide for setting up upper and lower output values thresholds, received after linear transformation from the current slot values.

Gamma correction Ch *field***,** located on the right, enables to set gamma-correction separately for each slot

Gamma correction *field*, located on the left, enables to set gamma-correction for the entire created image (gamma-correction curve dialog is shown in the rectangular in right bottom corner).

Notably, transformations are done in the order specified above for the controls.

Button **Preview** enables to preview the image, without getting out of the dialog (in case of a new image creation rather than modification of the existing, this button is not available).

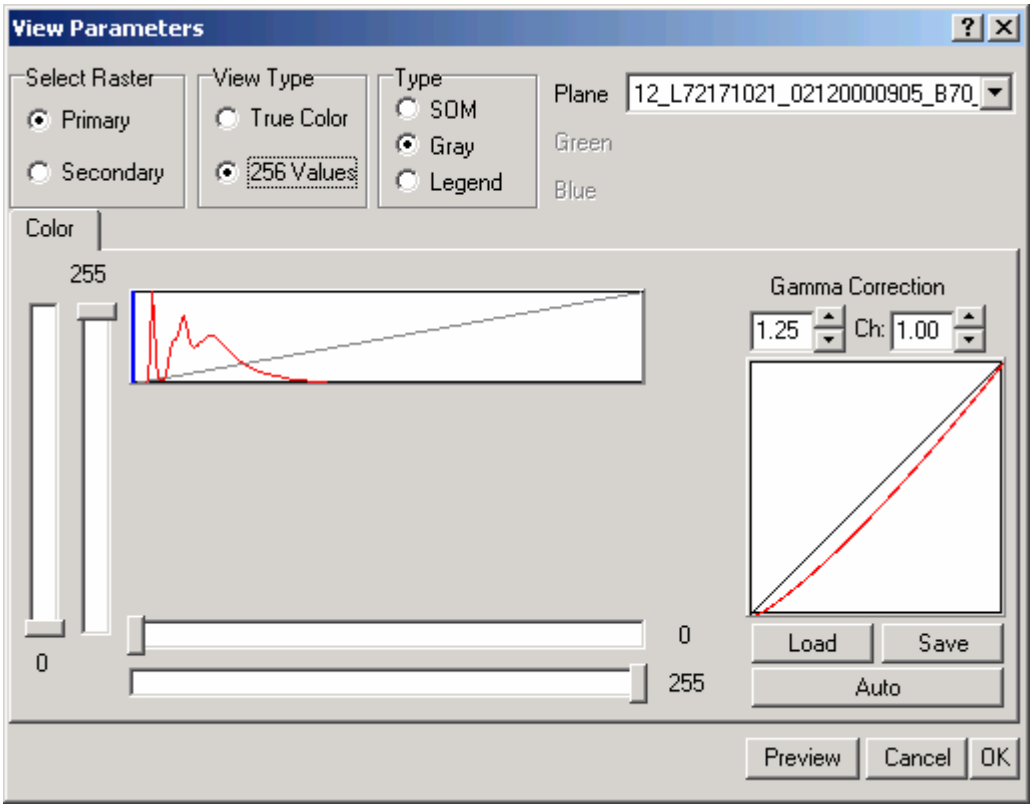
Button **Load** enables to load image parameters (band and gamma-correction sliders position) from a file.

Button **Save** enables to save image parameters (band and gamma-correction sliders position) to a file.

Button **Auto** performs image auto contrasting.

Creating monochrome display

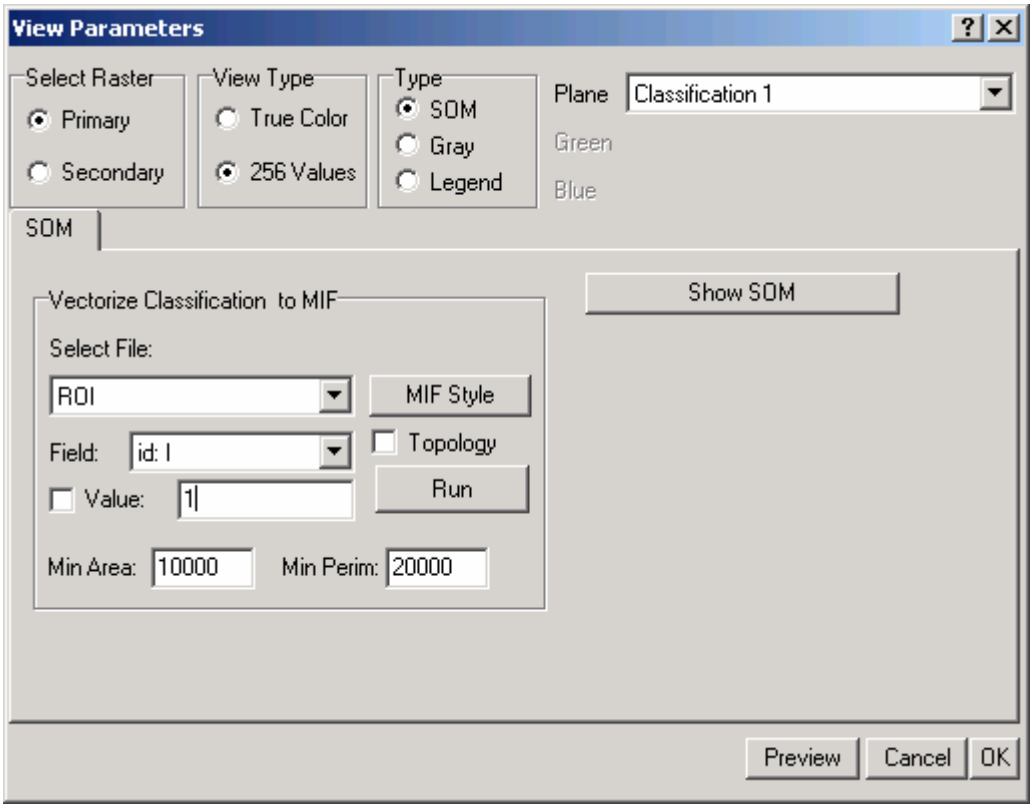
Monochrome display dialog is shown below:



Unlike the aforementioned dialog, this dialog has only one slot, therefore there is no control for the current slot selection.

Creating image using neural net classification results

When displaying the image based on the slot values, containing the classification results, the dialog looks as follows:



Besides the abovementioned controls, the dialog has the group of classification results vectorizing controls and Show SOM button, meant for palette control. It should be noted that all controls, designed for slot values transformation, are in the unit transformation status and inhibited, which is important for correct classes numbers interpretation.

Button **Show SOM** enables to open palette creation dialog using the trained neural network.

List box **Select File** enables to select vector layer, where the vectorized classification results will be put (classes selected in the palette creation dialog using the trained neural network, are subject for vectorizing).

List box **Field** enables to select the vector layer attributive table field, where the vectorized objects code will be save to (field must have integer numbers).

Checkbox **Topology** – if checked, partial topology is built prior to saving vector objects (in one object is inside the other bigger one, an island is "cut out" in the outer object). This operation takes some time, however it is convenient for further vector objects processing (area calculation, etc.).

Checkbox **Value** controls the code saving (the code is saved only if this control is checked).

Field **Value** enables to display vectorized objects code.

Field **Min Area** enables to specify minimal area of vectorized objects.

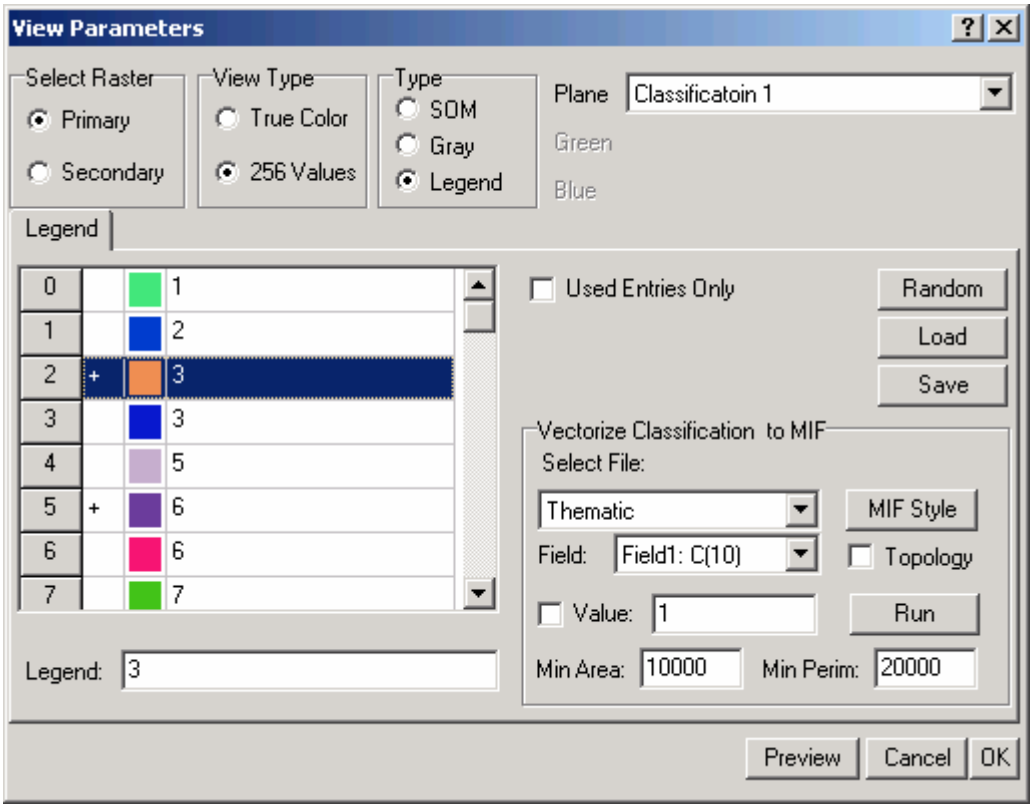
Field **Min Perim** enables to specify minimal perimeter of vectorized objects.

Button **MIF Style** enables to specify drawing style of the vectorized objects.

Button **Run** initiates vectorization.

Creating image using legend

When displaying the image based on the slots values, containing classification results, using the legend, the dialog looks as follows:



List box **Plane** enables to select slot, containing the classification results.

Table **Legend** contains the list of all values within the legend, indicating class code, vectorization feature, output color and comments for each value (legend class).

Checkbox **Used Entries Only** – if checked, only codes of the classes, present in the corresponding slot, are put in the legend.

Button **Random** enables to select random colors for classes display.

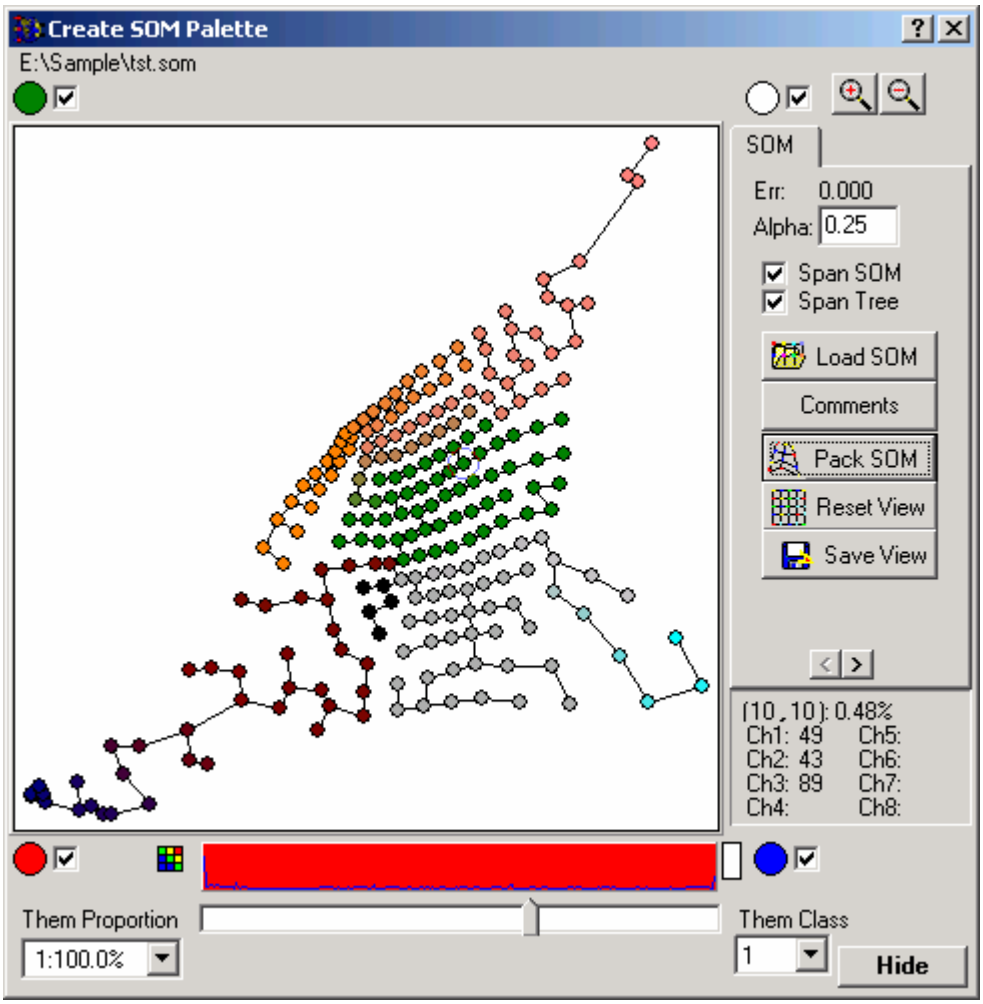
Button **Load** enables to load previously saved legend from file.

Button **Save** enables to save prepared legend into file.

Field **Legend** enables to add comments to the corresponding legend class.

Vectorize Classification to MIF *group* enables to transform classes selected for vectorization into vector objects in the same way as during neural net classes' vectorization in the relevant dialog, described above.

***Creating maps based on neural net classification results (Create SOM Palette** dialog)*

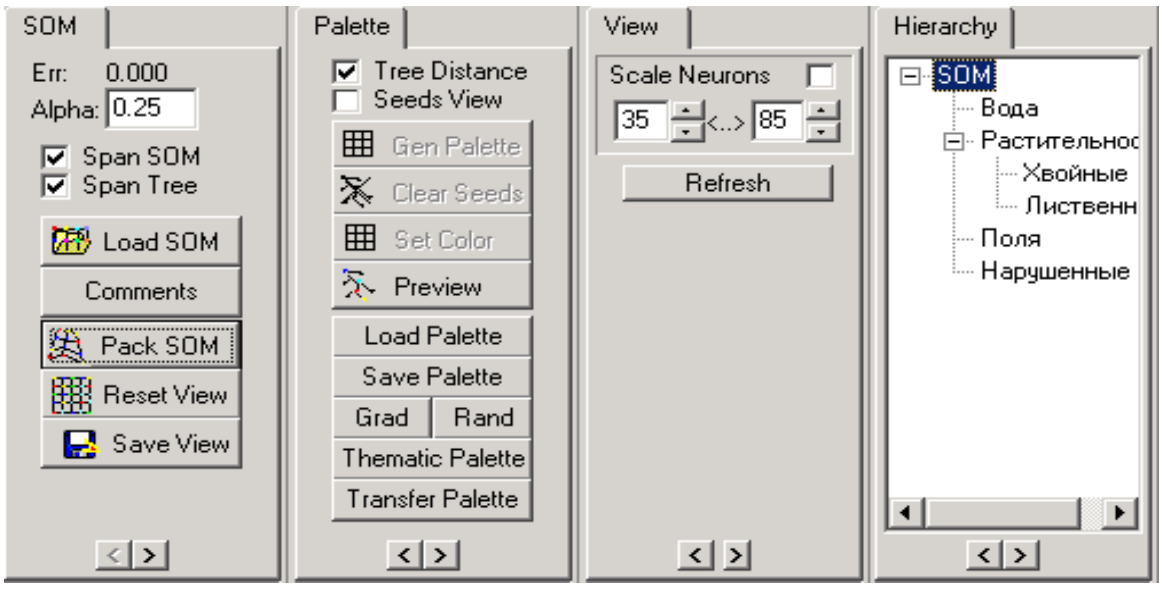


Classification results display control

The central part of the dialog is taken by a square, displaying the loaded trained Kohonen neural net.

Zoom in and zoom out buttons in the top right corner enable to make the view larger or smaller.

The area of *tabs* controls is located to the right of the square, containing data to select the required classes and palette for their display according to the thematic task to be resolved.



- **SOM** *tab* loads the trained Kohonen neural net, builds two options of minimal backbone tree, as well as displays Sammon's classes, represented by the net neurons, their visualization and export in form of graphic files.
- **Palette** *tab* creates thematic palette according to the task to be resolved.
- **View** *tab* saves the data, contained in the Kohonen neural net, compressed in two primary raster slots.
- **Hierarchy** *tab* creates classes' hierarchy in neural net.

Buttons with < and > symbols enable to move from one tab to other.

Color gradient palette creation controls are placed in all four corners of the rectangular. Each control has a color circle. Standard MS Windows color selection dialog can be opened by clicking mouse button on each of them, as well as the *checkbox*, indicating the requirement to use this color for the palette creation.

On the right, under the *Tabs* controls area, there is a data field, displaying the current net neuron position in its topological grid and the values of the first eight elements of the class center vector, corresponding to this neuron.

Building thematic palette and vectorization preparation

Controls specifying thematic data, contained in the calibrated Kohonen neural net, are located in the bottom of the dialog.

List box **Them Class**, located on the right, contains the list of all thematic classes, used during the trained neural net calibration, and enables to select the one, which the thematic legend is being built for.

Selection of the current thematic class defines the image view, built in the rectangular. It displays the total number of pixels (blue curve) allocated to the classes during calibration and thematic proportion (red histogram) for all neural net classes, arranged according to the share of the thematic class selected.

List box **Them Proportion**, located on the left, contains the proportion of all thematic classes within the class, corresponding to the current neuron (current neuron can be selected using left mouse button).

Slider bar in the bottom of the dialog enables to select the "threshold" class: all classes, matching with neurons and located to the right of the threshold, are considered as belonging to the selected thematic class, whereas the ones located to the left – as not belonging.

Color rectangular to the right of it enables to create the color of neurons, referred to the selected thematic classes (other neurons receive colors, corresponding to the RGB composition of the first three vector elements of relevant classes).

Thematic palette composition is done by clicking left mouse button on the color rectangular . All neurons referred to the selected thematic class become outlined and can be used for vectorization as a result.

Neuron selection mark can be changes using standard MS Windows procedure (with **Shift** key pressed, click left mouse button on the relevant neuron).

If left mouse button is clicked with pressed **Alt** key, the class will be selected in yellow (the map should already be visible).

SOM tab

As it was already mentioned, the main purpose of this control tab is to geometrically map the thematic data, contained in the neural net.

Field **Alpha** specifies the parameters value, used in Sammon's map creation iteration.

Field **Err** specifies the current Sammon's map error.

Checkbox **Span SOM** enables to specify the requirement to build and view minimal backbone tree, corresponding to the entire classes distance matrix.

Checkbox **Span Tree** indicates the requirement to build the minimal backbone tree on the distance matrix in neural net topological grid, rather than on the entire classes distance matrix.

Button **Load SOM** loads the trained Kohonen neural net from file.

Button **Comments** enables to view loaded neural net parameters.

Button **Pack SOM** enables to execute one Sammon's map creation iteration.

Button **Reset View** enables to use neurons coordinates in topological grid instead of Sammon's map.

Button **Save View** enables to save the built neural net image as a Windows meta-file.

Palette tab

This control tab is meant for the thematic palette selection, using the following controls:

Checkbox **Seeds View** specifies the output color in the circles, corresponding to neurons: if checked, "color seeds" of the future palette are displayed, if unchecked – the palette itself.

Button **Gen Palette** enables to build the palette using "color seeds" by interpolation according to the neurons coordinates within the **SOM** area.

Button Clear **Seeds** deletes all "color seeds".

Button **Set Color** changes the color of all currently selected neurons.

Button **Preview** enables to preview the palette results in the map window.

Button Save **Palette** saves the built palette (including the neurons position within the **SOM** window) to a disk file.

Button **Load Palette** loads previously built palette (including neurons position within the **SOM** window) from a file.

Button **Grad** enables to create the gradient palette, using **Color** controls (located in the **SOM** window corners) as "color seeds".

Checkbox **Tree Distance –** if checked, (colors are interpolated using bilinear and linear methods, depending on the "seeds" number, according to the neurons position in the **SOM** window).

Button **Rand** enables to create random palette.

Button **Thematic Palette** enables to create the palette, where each thematic class has its color; thematic classes are numbered, starting 1 in the same order as they are presented in the thematic classes' list (located to the right of the slider bar in the bottom of the dialog). This palette type is meant for remote sensing data thematic interpretation visualization.

View tab

Checkbox **Scale Neurons** enables to change the neurons display scale depending on the number of pixels related to the specific neuron.

Minimal and maximal size (the value, corresponding to maximum and minimum of pixel related to this class) is specified in percents.

Hierarchy tab

Neural net classes' hierarchy provides for storage of data about thematic inclusion of one group of classes into another and, for example, can be the basis for the creation of hierarchical legend. Besides, the hierarchy is the basis for the creation of the corresponding neural nets hierarchy, enabling to the make a detailed research of this or that group of classes.

Current neural net is recorded in the node of hierarchy and cannot be changed. When loading the hierarchy tree from a file, the neural net, stored in the hierarchy node, becomes current (as if it is loaded from a file into the **SOM** window).

If current neural net size and the replacing neural net size from the hierarchy node are different, the error message is popped up and the hierarchy tree would not load.

To refer neural net neurons to this hierarchy node, they should be selected (by clicking right mouse button on them with pressed **Shift** key) and then dragged to the corresponding hierarchy node, using the "drag-and-drop" Windows tool.

The same mechanism is used to drag neural net neurons from one hierarchy node to another (by clicking left mouse button on the field of the relevant initial node and dragging it to the new node).

To build and change the hierarchy tree itself, use **Pop Up** menu, opened by clicking the right mouse button on the hierarchy tab field:

Hierarchy tab Pop Up-menu

Add	
Node	Add new node to the hierarchy at the current level
Child	Add child node of the hierarchy at the current level
Delete	

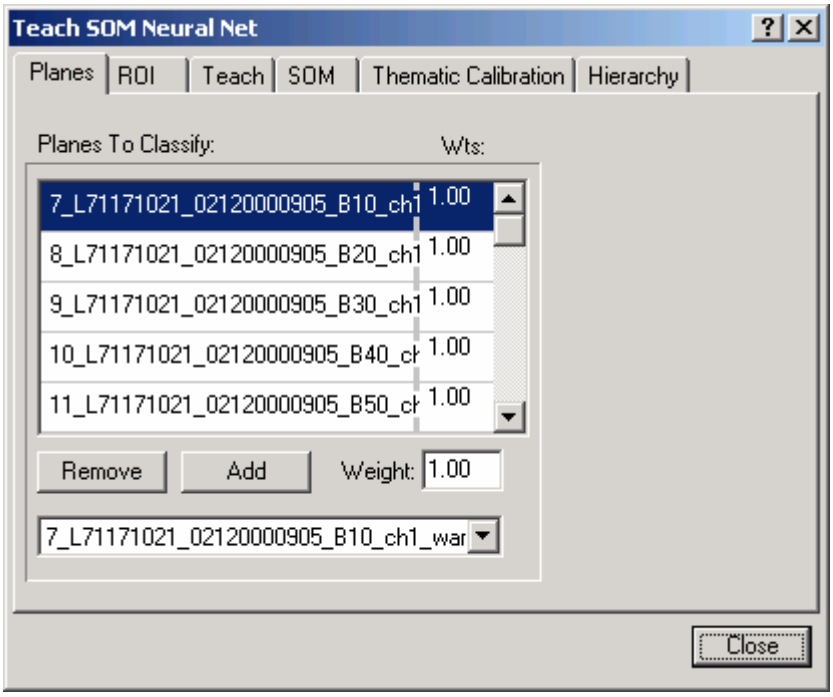
Node	Delete current node of hierarchy (all childs included)
Child	Delete all child nodes of current hierarchy level
Select	Select all neurons within the hierarchy node in subject (enables, for example, to set uniform color legend for all neurons, referred to the hierarchy node)
Load tree	Load hierarchy tree
Save tree	Save hierarchy tree

***Kohonen neural net training & calibration (Teach SOM Neural Net** dialog)*

This option is opened by selecting **Teach SOM** item in **Classification** menu and is designed for neural network training control.

This dialog is a multi-tab interface containing all required tools for the creation, training and calibration of Kohonen neural nets according to the specified raster slots of the primary raster.

Planes *tab* enables to select slots and their pre-processing method:



This tab contains controls, setting the number and order of primary raster slots used, as well as the normalization method.

Table **Planes to Classify** contains the list of primary raster slots, involved in training and having two columns each:

- Column describing primary raster slot.
- Column with slot weights, multiplied by the values prior to vector input in the neural net, thus changing the impact of the relevant slot on the training results.

To control normalization, click left mouse button on the corresponding cell; to change weights, use the relevant field in the tab bottom.

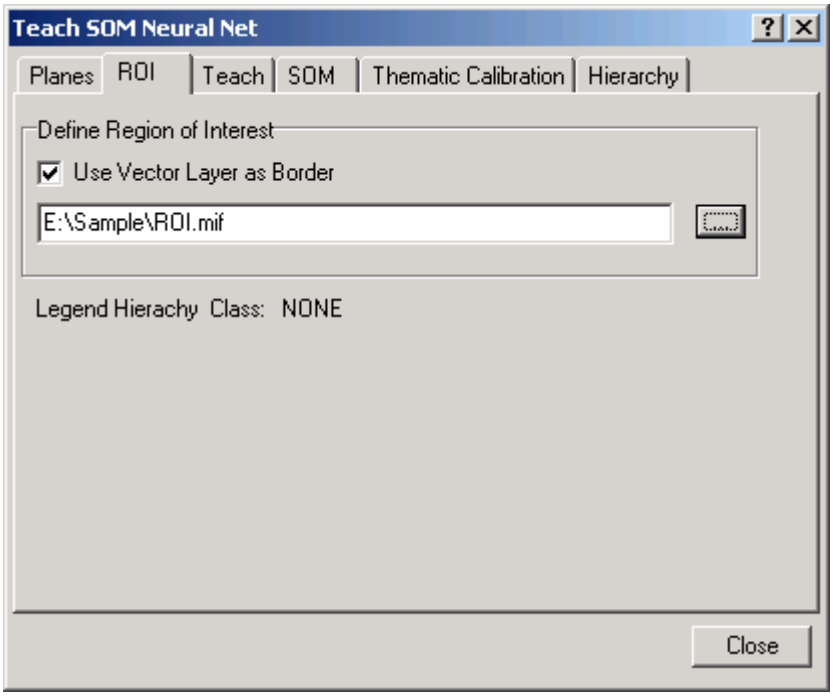
Button **Remove** deletes current slot from the list for training.

Button **Add** adds a new slot in the list for training.

Field **Weight** changes the plane weight.

List box **Plane** below enables to change current slot in the list for one of the primary raster slots.

ROI *tab* enables to define the region of interest (mask), which pixels are used for neural net training:



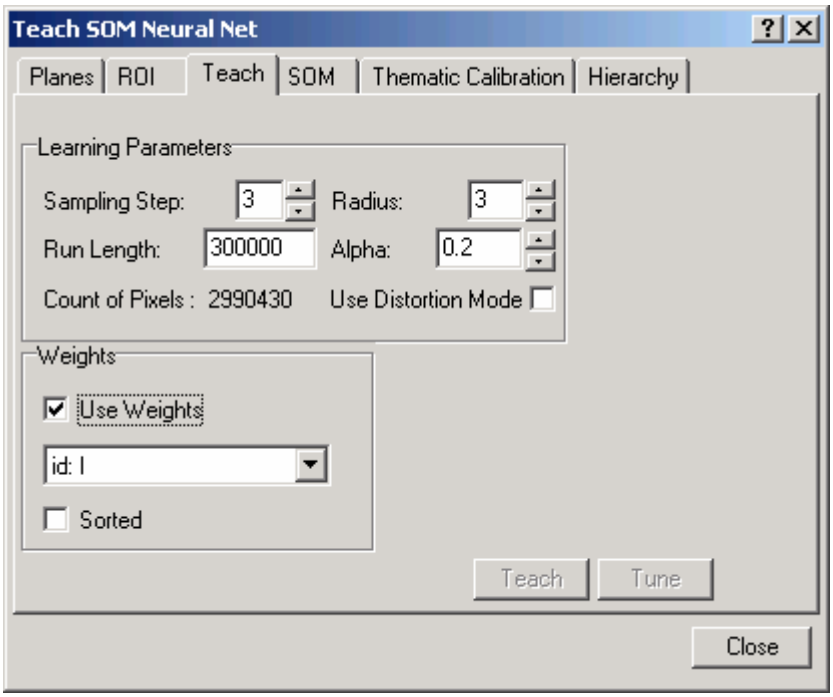
The application provides for two selection options: using vector map in MIF format as mask and using the slot previously classified by another neural net (by specifying hierarchy nodes, containing the required classes; in this case mask control is done via the hierarchy tab).

Checkbox **Use Vector Layer as Border** indicates the requirement to limit the training area by the regions, contained in vector layer file.

File selection push-button enables to select the vector map file, containing training area borders.

Slots names and hierarchy nodes, used for mask creation, are indicated below. If both vector map and hierarchy mask are applied, only pixels within both masks are used for training.

Teach *tab* enables to specify training parameters:



Field **Sampling Step –** pixels sampling step during training and calibration (value 2 indicates the use of each second pixel in row)

Field **Radius** – neighborhood radius value, used in training.

Field **Run Length** – total number of used pixels (if the region being trained contains fewer pixels, the latter are used several times).

Field **Alpha** –initial value of training rate.

Checkbox **Use Distortion Mode** enables to change training mode (used if there was no self-organization of neural net or classes' ordination during regular training).

Button **Teach** initiates neural net training.

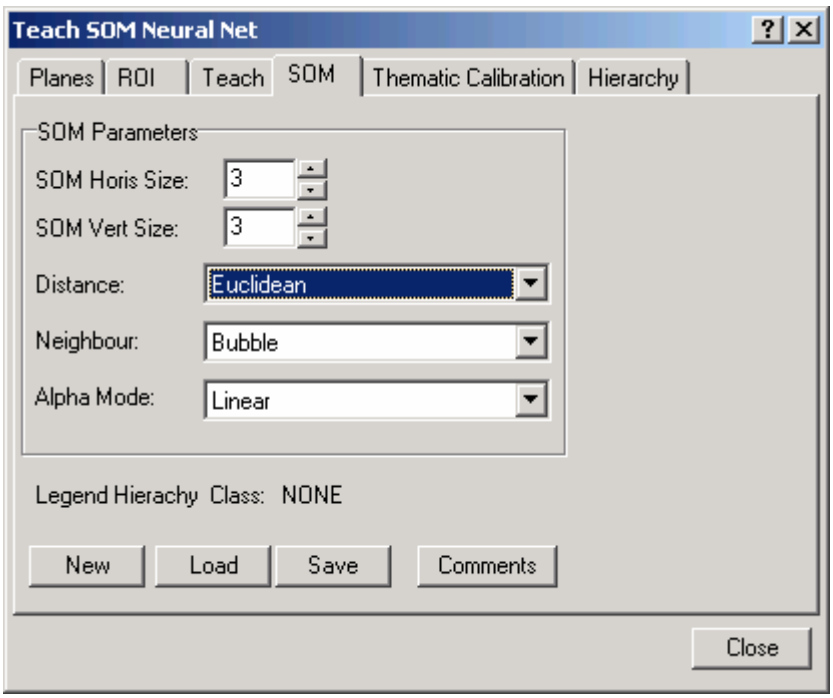
Button **Tune** initiates "tuning" of reference elements of neural net classes.

Checkbox **Use Weights** indicates the requirement to use weights during neural net training, with reference to the vector map objects. The bigger is the object weight, the stronger is the influence of the corresponding image area on the neural net training results (neural net thematic orientation).

List box **Use Weights** contains the list of all attributive table fields of the used vector map and enables to select the one, which contains weights (the field must be in integer or floating point format with positive values).

Checkbox **Sorted** indicates the requirement to sort out vector map objects based on weights values (used for training thematic attitude).

SOM *tab* provides for the neural net parameters setting:



Field **SOM Horis Size** specifies neural net grid horizontal size.

Field **SOM Vert Size** specifies neural net grid vertical size.

List box **Distance** specifies distance calculation method:

- **CityBlock -** L1 metrics
- **Euclidean –** Euclidean distance
- **Spectral Angle** – spectral angle (calculated as relation of scalar vectors product to the product of their lengths)

List box **Neighbour** specifies the way to define neuron's neighbors in the grid:

- **Gaussian –** Gaussian neighborhood
- **Bubble -** "Bubble" neighborhood

List box **Alpha Mode** specifies the way to change the training rate:

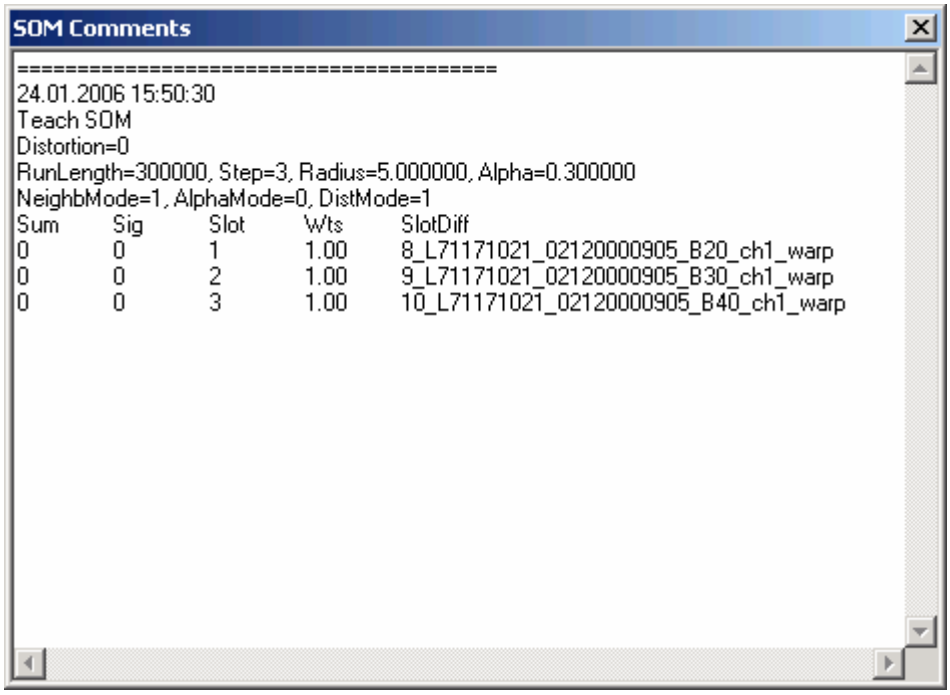
- **Invers Time** – in inverse proportion to the iteration number
- **Linear -** linear

Button **New** creates a new neural net with the specified parameters.

Button **Load** loads neural net from file.

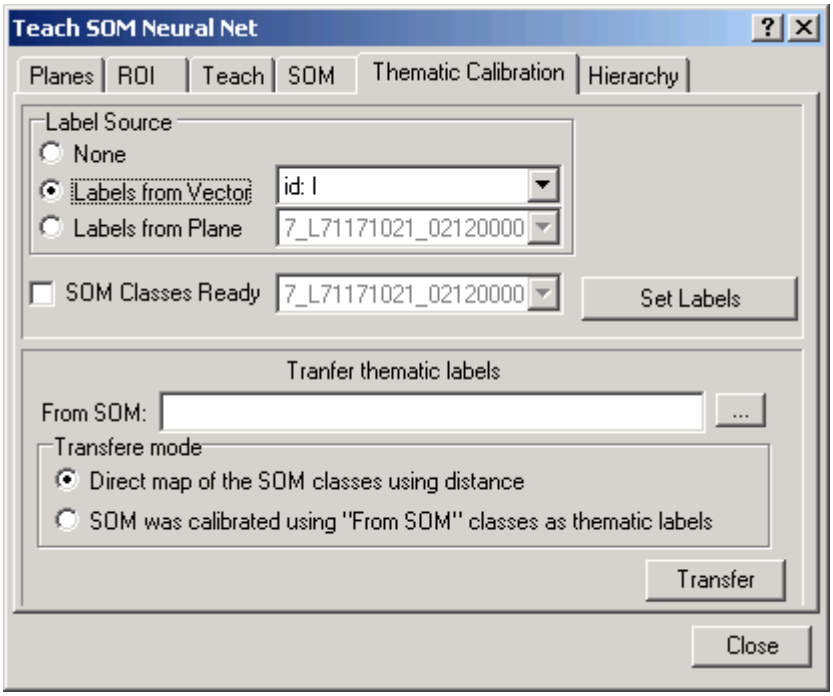
Button **Save** saves neural net to file.

Button **Comments** enables to create and edit comments, stored for the neural net. In particular, parameters, used during the neural net training and calibration, are stored here, including the slots composition and normalization method. Standard multi-line edit window appears on the screen after pressing this button as shown below. The user can add his own comments, which are stored into file when neural network is recorded.



Information about the reference of the current neural net to a certain hierarchy node is available in the bottom of the tab (neural net can be taken from the hierarchy node for additional training or saved in the hierarchy node).

Thematic Calibration *tab* provides for the neural net thematic calibration:



Group **Label Source.** Three options are possible:

- **None** deactivates thematic labels.
- **Labels from Vector** enables to select the attributive data field for thematic labels use. Vector layer, specified in the second tab of the dialog, is used for calibration.
- **Labels from Plane** enables to use thematic labels from raster plane (raster map or classification results).

Checkbox **SOM Classes Ready** enables to set new thematic labels on the previously classified layer (the layer is specified in the list on the right).

Button **Set Labels** – initiates thematic calibration of the previously trained neural net.

Field **Tranfer thematic labels (from SOM)** specifies the neural net, where the data is taken from and provides for the data transfer from one grid (target grid) to another (source grid – loaded in SOM tab).

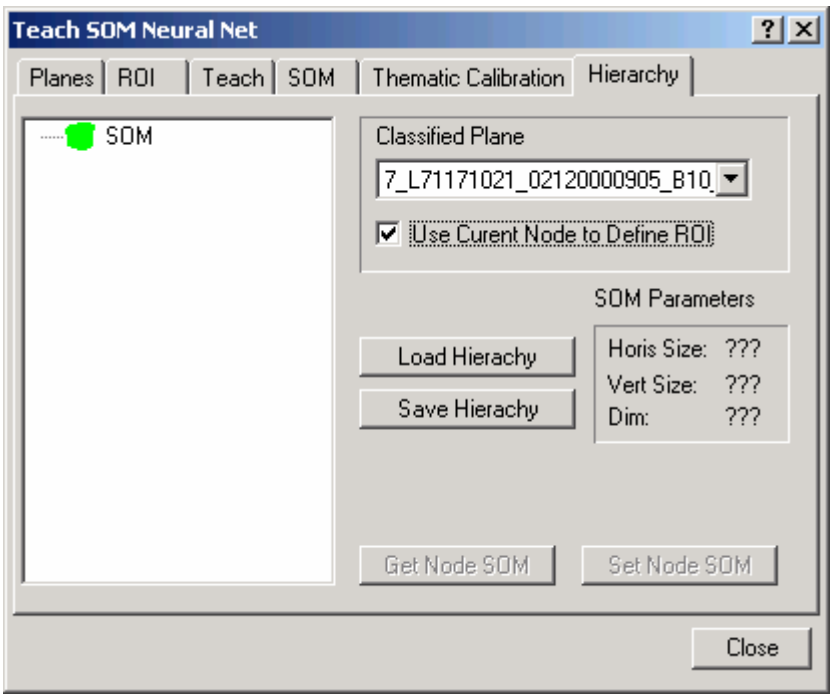
Group **Transfere mode.** Two options are possible:

- Source grid classes' centers are used as data and the nearest neuron (class) is searched for them in the target grid, used for calibration.
- The layer, classified by the target grid, is used (this layer contains the numbers of this grid classes), and it was used for the source grid calibration.

In this way, each neuron of the source grid is matched with the target grid neurons distribution, whereas each target grid neuron is matched in its turn with the thematic classes' distribution.

Button **Transfer** initiates thematic labels transfer.

Hierarchy *tab* provides for the creation and operation with neural net hierarchy, corresponding to the previously created hierarchy of classes. Neural net is on the node of this hierarchy. Subsets of these neural net classes are below, with all classes of subsidiary hierarchy nodes always on the nodes of the higher level. Neural net can be saved for each hierarchy node. Such neural net is stored in the hierarchy file and can be used for further classification. Therefore, hierarchy is a convenience means of storing data about the hierarchy legend of classification and corresponding neural networks.



List box **Classified Plane** indicates the primary raster slot, containing neural net classification results, located on hierarchy node.

Checkbox **Use Curent Node to Define ROI** indicates that the classes from current hierarchy node will be used as mask during neural net training (only pixels, related to the mask, are used in neural net training).

Button **Load Hierarchy** loads hierarchy from disk.

Button **Save Hierarchy** saves hierarchy to file.

Button **Get Node SOM** enables to get the neural net from the hierarchy node and make it current (available for training and calibration) the same way as if it were loaded from disk or created in the **SOM** tab. Notably, the current neural net can be saved into a separate file.

Button **Set Node SOM** enables to save current neural net (created or loaded from a separate file using **SOM** tab controls) into the relevant hierarchy node. Hierarchy node, corresponding to the current neural net (where the neural net was taken from or saved to) is highlighted with the proper sign.

Below are the possible "painting" options for hierarchy nodes:

	SOM is stored in hierarchy node, which is not currently used
	No SOM in hierarchy node, however classes, assigned to the node, are used for masking
	SOM is saved in hierarchy node, which is not currently used, however classes, assigned to the node, are used for masking

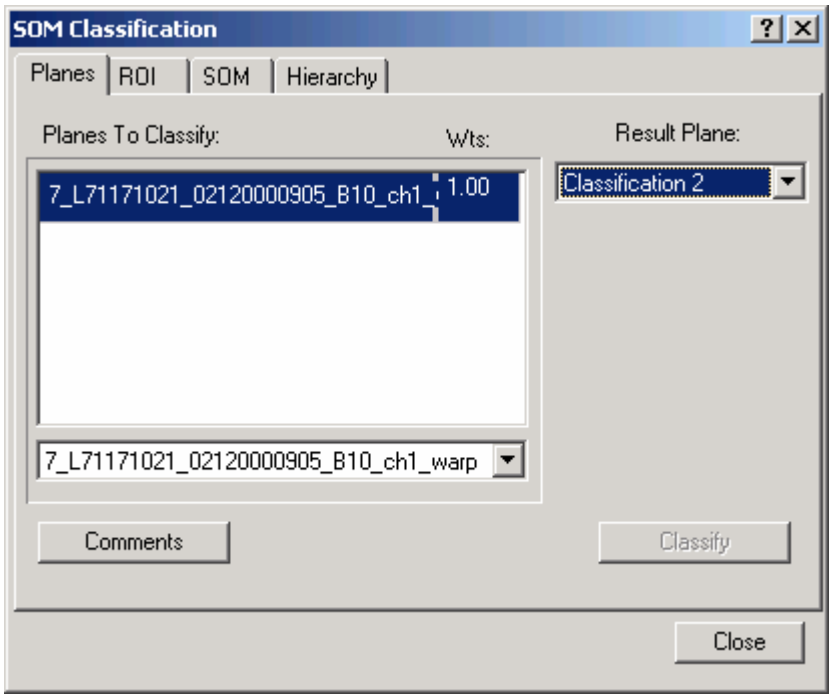
	SOM is saved in hierarchy node, which was either made current (taken from the hierarchy node) or saved in hierarchy node
	SOM is saved in hierarchy node, which was either made current (taken from the hierarchy node) or saved in hierarchy node, however classes, assigned to the node, are used for masking
	No neural net in hierarchy node and classes in the node are not used for masking

***Classification using Kohonen neural networks (SOM Classification** dialog)*

This dialog can be opened by selecting **SOM Classification** item in **Classification** menu and is designed for the classification control using previously trained neural net and primary raster slots.

This dialog has several tabs and contains all the required tools for the classification of the specified set of primary raster slots.

Planes *tab* enables to select slots and their pre-processing method:



This dialog contains controls, specifying raster slots used in classification.

Table **Planes to Classify** contains the list of primary raster slots used in training. Each of them has two columns:

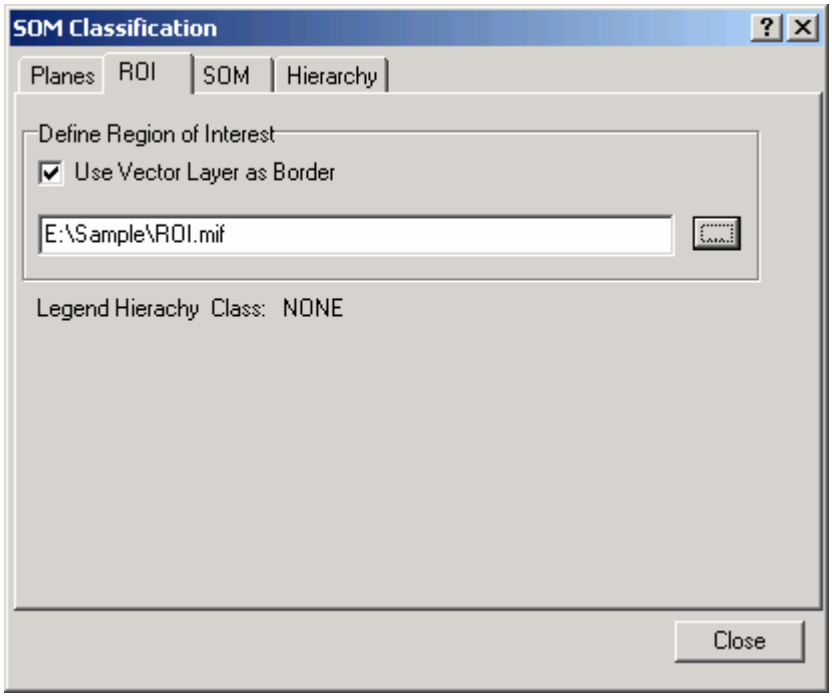
- Column describing primary raster slot.
- Column with slot weights, multiplied by the values prior to vector input in the neural net, thus changing the impact of the relevant slot on the training results.

List box **Plane** below enables to change the current slot in the list for one of the primary raster slots.

Number of slots, normalization and weight parameters are taken from the comments file of the trained neural net.

List box **Result Plane** enables to select the slot for the classification results input.

ROI *tab* enables to select the part of the area (mask), which pixels are used for neural net training:



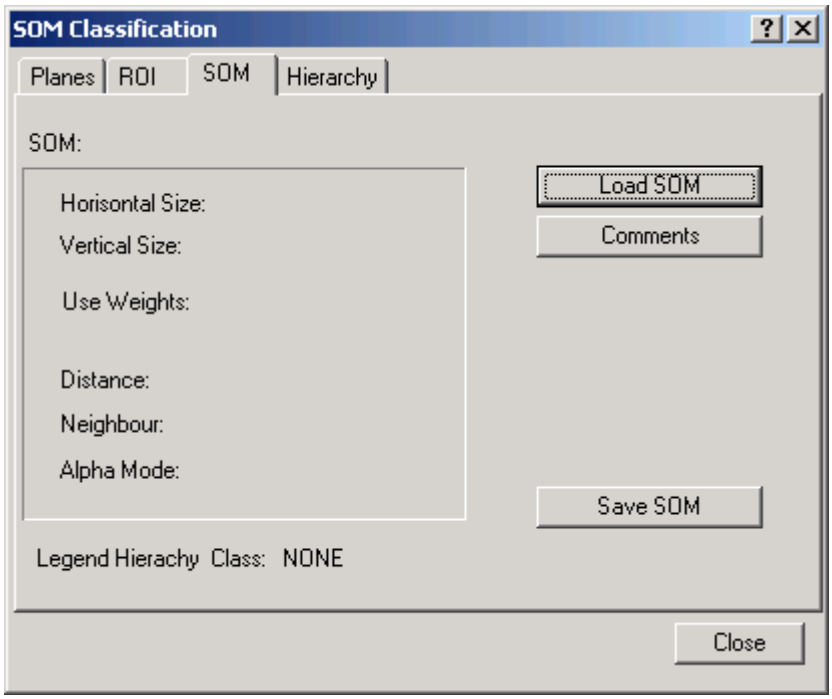
The application provides for two selection options: using vector map in MIF format as mask and using the slot previously classified by another neural net (by specifying hierarchy nodes, containing the required classes; in this case mask control is done via the hierarchy tab).

Checkbox **Use Vector Layer as Border** indicates the requirement to limit the training area by the regions, contained in vector layer file.

File selection push-button enables to select the vector map file, containing training area borders.

Slots names and hierarchy nodes, used for mask creation, are indicated below. If both vector map and hierarchy mask are applied, only pixels within both masks are used for training.

SOM *tab* enables to display neural net parameters and to load the net from file (if the neural net, stored in the hierarchy node, is not used):

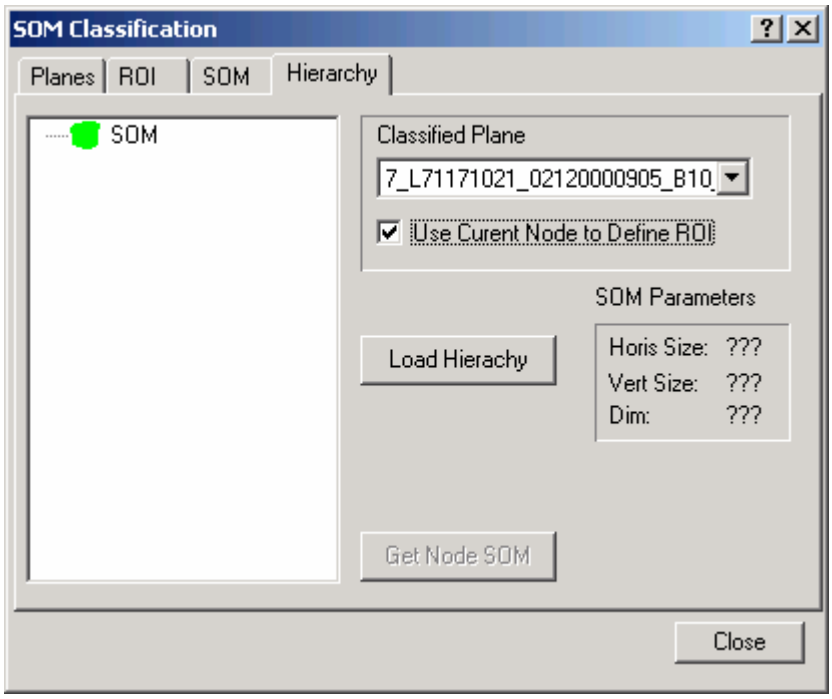


Button **Load SOM** loads the neural net from file.

Button **Comments** enables to view neural net comments.

Information about the reference of the current neural net to a certain hierarchy node is available in the bottom of the tab.

Hierarchy *tab* enables to control the neural net hierarchy, corresponding to the previously created hierarchy of classes. Neural net is on the node of this hierarchy.



List box **Classified Plane** indicates the primary raster slot, containing neural net classification results, located on hierarchy node.

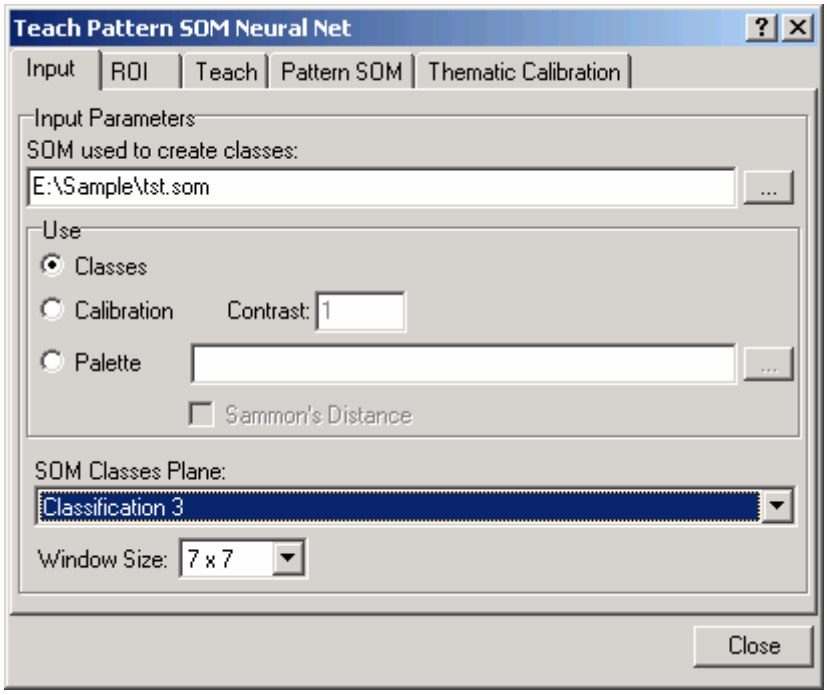
Checkbox **Use Curent Node to Define ROI** indicates that the classes from current hierarchy node will be used as mask during neural net training.

Button **Load Hierarchy** loads hierarchy from disk.

Button **Get Node SOM** enables to get the neural net from the hierarchy node and make it current (available for training and calibration) the same way as if it were loaded from disk or created in the **SOM** tab. Notably, the current neural net can be saved into a separate file.

***Contextual Kohonen neural network training and calibration (Teach Pattern SOM Neural Net** dialog)*

This dialog can be opened by selecting **Teach Pattern SOM** item in **Classification** menu and is designed to control contextual neural net training process.



***Input** tab* controls:

Field **SOM used to create classes** specifies the neural net for classification.

Group of radio-buttons **Use:**

- **Classes** radio button enables to use histogram of classes from the slot, indicated in the list below, as input data for pattern **SOM**. Resulting pattern **SOM** input data vector length equals to the number of classes in **SOM**, used for classification of the slot indicated in the list, and contains counts for corresponding classes;
- **Calibration** radio button. If checked, histogram of thematic classes is used as input data, corresponding to the classes numbers, saved in the layer pixel from the list below (used **SOM** must be thematically calibrated). Created pattern **SOM** input data vector length equals to the number of thematic classes, and contains counts for corresponding thematic classes (extracted from the SOM, used to classify layer);
- **Palette** radio button. If checked, histogram of distances within **RGB** space from the given class color in the slot, indicated in the list below, to "color seeds" in the loaded palette is used as input data. Created pattern **SOM** input data vector length equals to the number of color seeds in the palette, and contains contributions of seeds into the class color.

Checkbox **Sammon's Distance** – if checked, distance to neurons with color seeds in the "packed" ordination plane is used instead of RGB distance (in Sammon's map).

List box **SOM Classes Plane** specifies the primary raster plane, classified using neural net (**SOM**), which will be used for pattern **SOM** neural net processing.

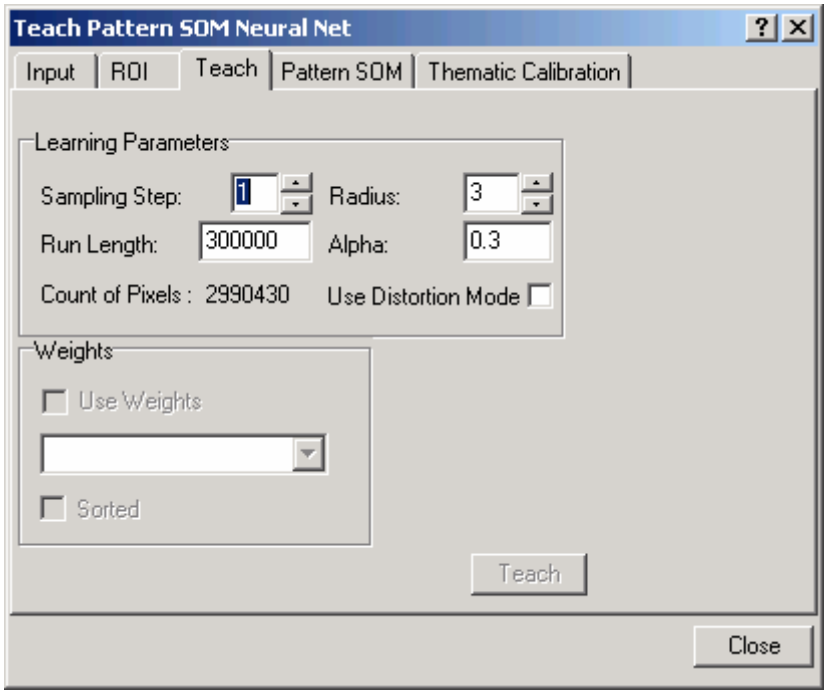
Field **Window Size** specifies the size of window.

***ROI** tab* controls:

Group **Define Region of Interest** indicates the requirement to limit the training area by the regions, contained in vector map file.

File selection push-button enables to select the vector map file, containing training area borders.

***Teach** tab* controls:



Field **Sampling Step –** pixels sampling step during training and calibration (value 2 indicates the use of each second pixel in row)

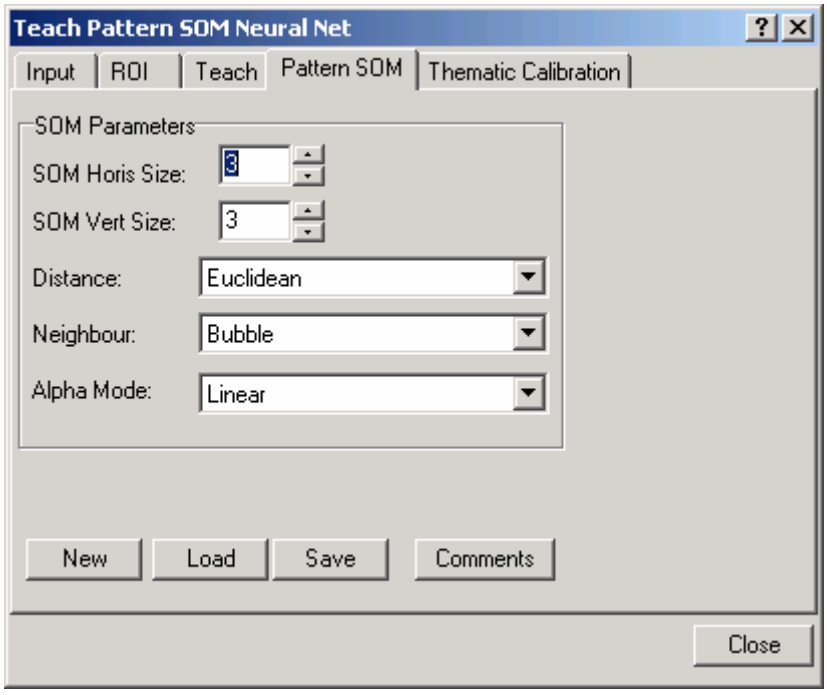
Field **Radius** – neighborhood radius value, used in training.

Field **Run Length** – total number of used pixels (if the region being trained contains fewer pixels, the latter are used several times).

Field **Alpha** –initial value of training rate.

Checkbox ***Use Distortion Mode*** enables to change training mode (used if there was no self-organization of neural net or classes' ordination during regular training).

***Pattern SOM** neural net tab* controls*:*



This tab contains neural net creation parameters.

Field **SOM Horis Size** specifies neural net grid horizontal size.

Field **SOM Vert Size** specifies neural net grid vertical size.

List box **Distance** specifies distance calculation method:

- **CityBlock -** L1 metrics
- **Euclidean –** Euclidean distance
- **Spectral Angle** – spectral angle (calculated as relation of scalar vectors product to the product of their lengths)

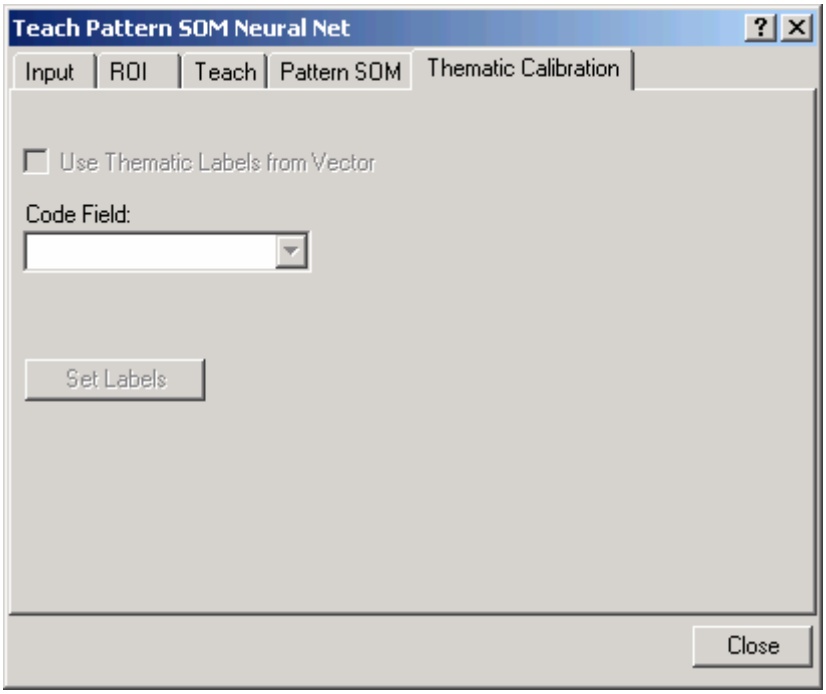
List box **Neighbour** specifies the way to define neuron's neighbors in the grid:

- **Gaussian –** Gaussian neighborhood
- **Bubble -** "Bubble" neighborhood

List box **Alpha Mode** specifies the way to change the training rate:

- **Invers Time** – in inverse proportion to the iteration number
- **Linear –** linear

***Thematic Calibration** tab* controls**:**



Checkbox **Use Thematic Labels from Vector** enables to select the vector map attributive data field (from **Code Field** *list box*) to use thematic labels. Vector layer, specified in the second tab of the dialog, is used for calibration.

Group of training and calibration control buttons

This group contains the following control buttons:

Button **Create** creates a new neural net with specified parameters.

Button **Load** loads neural net from file.

Button **Save** saves neural net to file.

Button **Comments** enable to create and edit comments, stored for neural net. In particular, parameters, used during the neural net training and calibration, are stored here, including the slots composition and normalization method. Standard multi-line edit window appears on the screen after pressing this button as shown below. The user can add his own comments, which are stored into file when neural network is recorded.

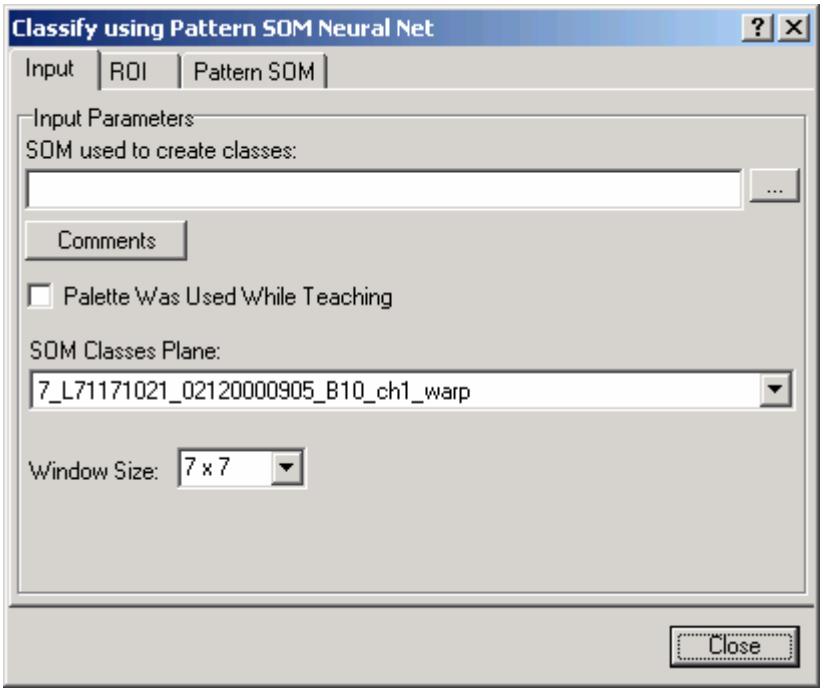
Button **Teach** initiates neural net training.

Button **Calibration** initiates calibration of the previously trained neural net.

***Contextual Kohonen neural network classification (Classify using Pattern SOM Neural Net** dialog)*

This dialog can be opened by selecting **Classify using Pattern SOM** item in **Classification** menu and is designed for classification control, using previously trained neural net and primary raster slots.

Input *tab* contains the group of input parameters for classification using pattern SOM:



Field **SOM used to create classes** specifies the neural net for classification.

Button **Comments** enables to view loaded neural net parameters.

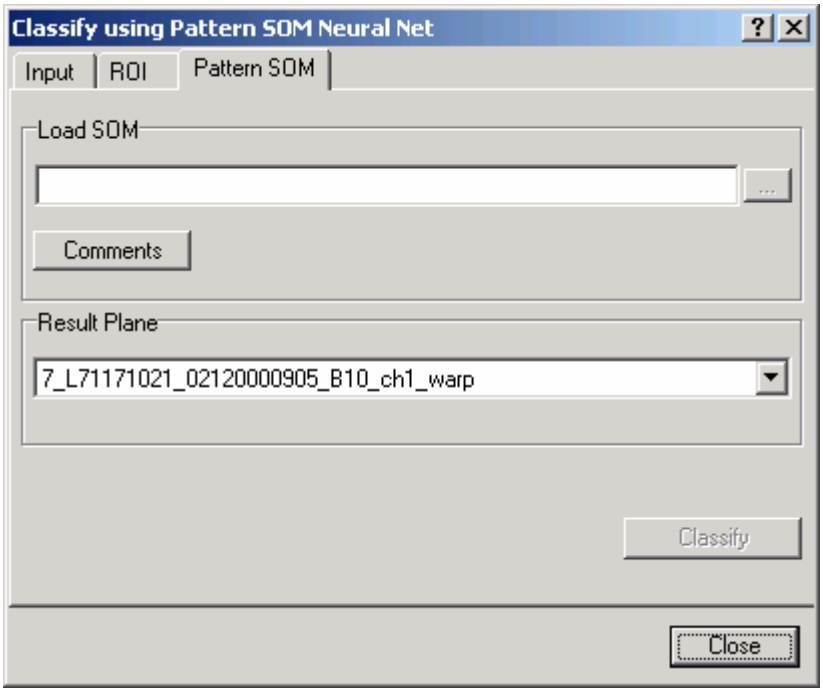
Checkbox **Palette Was Used While Teaching** must be checked if the palette was used in training.

List box **SOM Classes Plane** specifies the primary raster plane, classified using neural net (**SOM**), which will be used which will be used for pattern **SOM** neural net processing.

Field **Window Size** specifies the size of window.

ROI *tab* indicates the vector layer, limiting the training area (if *Checkbox* **Use Vector Layer as Border** is checked).

Pattern SOM *tab* contains the following controls:



Field **Load SOM** loads the neural net from file.

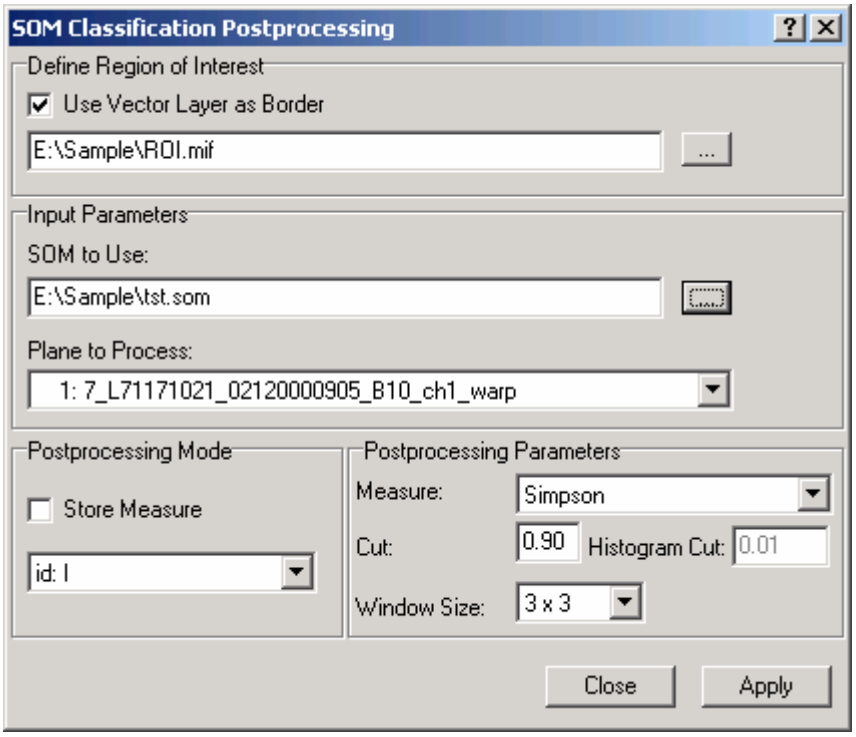
Button **Comments** enables to view parameters of loaded neural net.

List box **Result Plane** enables to select the plane for storing classification results.

Button **Classify** initiates the image classification.

***Classification results postprocessing using local window (SOM Classification Postprocessing** dialog)*

Classification results can be subject to postprocessing in local window in order to define thematic classes instead of neuron numbers. Select **Local Postprocessing** item in **Classify** menu. Postprocessing dialog is shown below:



SOM neural net classes' distribution (histograms in local window or areal object from the vector layer) is used as an input data vectors for postprocessing (comparing with the reference histograms). Reference histograms, corresponding to the thematic classes, are extracted from the neural net calibration. For example, if 4 thematic classes were used for the calibration of 3x3 neural net, containing 9 neurons, to each of them corresponds a histogram vector of 9 components, representing to the probabilities of 9 SOM classes for pixels from this thematic class.

Group **Define Region of Interest** controls area to be postprocessed.

Checkbox **Use Vector Layer as Border** indicates the requirement to restrict the image area being classified with the areal objects, contained in vector map file.

File selection push-button enables to select vector map file, containing region borders.

Group **Postprocessing Mode** controls postprocessing mode as shown below.

Check box **Store Measure.** Postprocessing can be done in two modes, depending on the status of the **Store Measure** *check box*: if it is not checked, each pixel value is replaced with the nearest (in the sense of histograms similarity) thematic class within the local window around this pixel. If the control is checked, the neural net classes' distribution is calculated for each areal object in the vector map file, and information on two closest thematic classes is stored in the selected attributive table field (which should be symbolic).

List box **Field** below enables to select the attributive table field of the indicated vector map (the field should symbolic with at least 15 symbols), where region thematic classification results will be recorded to.

Group **Input Parameters** controls the input data.

Field **SOM to use** contains the name of the previously trained and thematically calibrated neural net.

List box **Plane to Process** specifies the primary raster slot, containing classification results and subject for postprocessing.

Group **Postprocessing Parameters** controls the process.

List box **Measure** enables to select one of the agreement measures to compare SOM classes' composition using curent (window or object) classes histogram and reference histograms:

- **Simpson –** Simpson measure of agreement (use classes presence/absence only).
- **Fuzzy Simpson -** Simpson measure of agreement (use relative proportions).
- **Chi Square** – using p-value of Pearson Chi-square distributions agreement criteria.
- **Dot Product** – using scalar vector product (histograms are normalized in advance to unit length).
- **Euclidean (len=1)** – using Euclidean distance (histograms are normalized in advance to unit length).
- **Euclidean (sum=1)** – using Euclidean distance (histograms are normalized in advance to unit sum).
- **FuzzyART –** agreement measure based on the of adaptive resonance theory is used.
- **Correlation** – correlation coefficient is used.

All measures have values within the range from 0 (complete mismatch) to 1 (complete match of histograms).

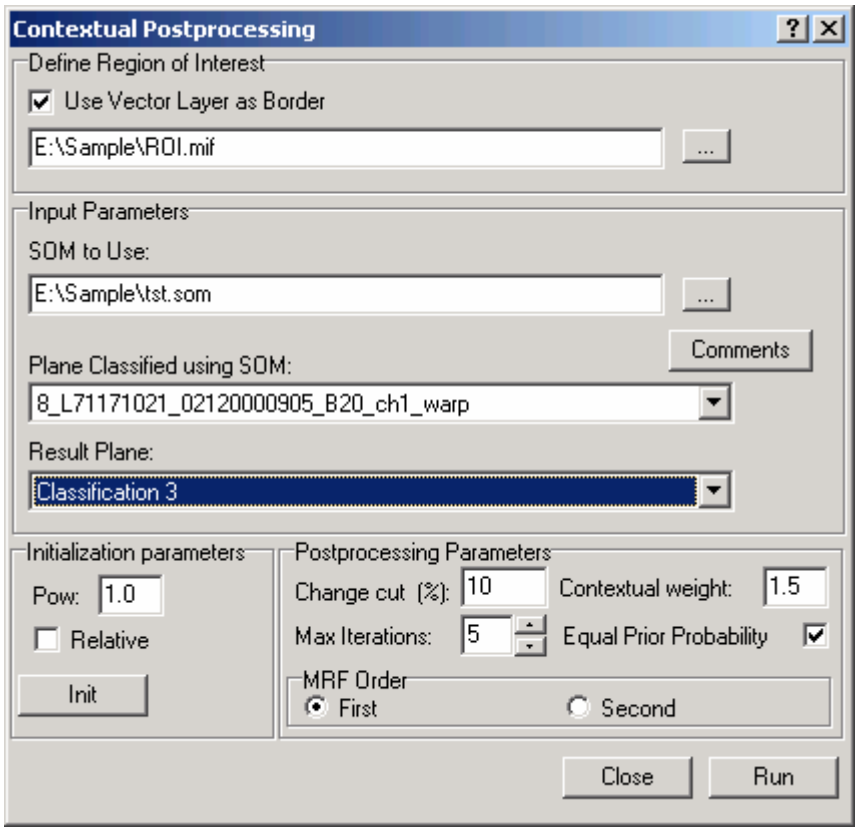
Field **Window size** specifies the size of square local window.

Field **Cut** specifies the threshold for the composition match measure. If measure values are beyond the threshold, thematic class is regarded as indefinite.

Field **Histogram Cut** is used only for Pirson criteria and specifies the minimal probability value, when several classes don't have to be combined in one for histogram calculations (this is done to improve the algorithm stability).

***Classification results postprocessing using Markov random fields (Contextual Postprocessing** dialog)*

Classification results can be subject to postprocessing in order to detect the thematic classes' distribution, best matching with neural net calibration, throughout the entire image being classified. Select **Contextual Postprocessing** item in **Classification** menu. Postprocessing dialog is illustrated below:



Thematic data, contained in the previously trained and calibrated Kohonen neural net, is used in this postprocessing mode. Based on this data, thematic classes' probability distribution is calculated for each pixel of the region being classified. This distribution is used both to specify the initial values of classes and to build the final classification.

Notably, already built classification can also be used as initial thematic classes' values, e.g. vector map thematic rasterization results, if this map is updated.

Received initial thematic classes' distribution is verified by iteration using contextual data to assign a thematic class to each pixel, best matching with both distribution of probability received from neural nets and with classification of this pixel "neighbors", with the help of Gibbs energy minimization while applying Markov random field.

Checkbox **Use Vector Layer as Border** indicates the requirement to restrict the image area being classified with the regions, contained in vector map file.

File selection push-button enables to select vector map file, containing region borders.

Field **SOM to use** contains the name of the previously trained and thematically calibrated neural net.

File selection push-button enables to select file, containing this neural net.

Button **Comments** enables to view the comments, contained in the neural net file.

List box **Plane Classified using SOM** enables to select the raster slot in the field, containing neural net classification, for which the selected neural net was used (this slot contains codes of neural net classes).

List box **Result Plane** enables to select the raster slot in the field, where the classification results have to be stored (codes of thematic classes).

Field **Change Cut** specifies the threshold value (percent in classification changes between iterations), which enables to assess the classification as stable and to suspend iterations.

Field **Max Iterations** specifies the maximum number of iterations.

Field **Contextual weight** specifies the degree of the given pixel thematic class compliance with the thematic classes of its neighbors (if the weight is 0, the neighbors have no impact on the final classification of this pixel; if the weight is big, only the initial distribution and neighbors can have an effect on the final classification).

Checkbox **Equal Prior Probability** enables to control the selection of initial values of classes based on their probability distribution. If checked, relative probability, rather than the absolute one, is used during the thematic classes' initialization (relation of this thematic class probability for this neuron to the prior probability of thematic class).

Group of radio-buttons **MRF Order** specifies the neighborhood order for the Markov random field (if the order is "second", the program runs a bit longer, giving more spatially "concurrent" results of thematic classification).

Field **Pow** contains probability "contrasting" index during the initialization: if its value is 0, the initial classes' distribution is independent of the probability distribution in the relevant net neuron. If the value is big (around 100) the thematic class, which probability is the maximum for this neuron, is used as initial value. Value 1 corresponds to the use of the existing probability distribution for the neuron (initial or relative) without modification.

Checkbox **Relative** enables to mandatory consider all prior thematic classes' probability as equal, regardless of calibration data in the neural net (this may be required, if thematic classes' distribution, used in calibration, does not correspond to their actual distribution for this area).

NeRIS application menu description

File		
Create primary raster		Create a new primary raster
Load primary raster		Load primary raster from file
Open secondary raster		Load secondary raster from file
Save primary raster		Save primary raster to file
Import		Import raster files to **NeRIS**
Export		
Exit		Exit **NeRIS**
View		
Display parameters		Change display parameters
Save image as …		
Image scale		
Set scale		
Fit to window		Fits all the image into the program window
Zoom in two times		**Zoom in the image twice**
Zoom out two times		**Zoom out the image twice**
Original scale		
Reset		
Vector		
Vector layers control		
Undo remove		
Delete object		
Create object		
Create point		
Create line		
Create polyline		
Create multi-line		
Create region		
Select object		
Unselect		
Set drawing style		
Set selection style		
Utilities		

Primary raster properties		Primary raster service commands (add, delete and clean slots, create blank vector map, slots importing)
Secondary raster properties		Secondary raster service commands
Patch primary raster		Open image mosaicking tools
Principal components		Open principal components calculation tools
Filtration		Open image filtration tools
Rasterize vector layer		
Statistic selection		
Change class number		
SOM utilities		
Classification		
Teach SOM		
SOM classification		
Teach Pattern SOM		
Classify using Pattern SOM		
Postprocessing Using Local Window		
Contextual Postprocessing		
Window		
Create working window		Create a new window for raster display
Synchronize windows		Synchronize the location of scrollers in all windows according to the current window (coordinates of all windows' centers are equal to the coordinates of the current window center)
Cascade		Cascade raster display windows
Horizontal		Arrange raster display windows horizontally
Vertical		Arrange raster display windows vertically
Close all windows		Close all windows
Refresh window		
Help		
CONTENTS		
MANUAL		
ABOUT THE PROGRAM …		About NeRIS application
Navigation		
Measure distance		Measure distance within the map window